UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Ingeniería en Informática

Proyecto Fin de Carrera

CREACIÓN DE MODELOS DE PREDICCIÓN ORIENTADOS A LAS APUESTAS EN EVENTOS DEPORTIVOS

Autor: D. José Ángel Vivar Angulo

Director: Prof. Agapito Ledezma Espino

Junio, 2010

A mis padres, Paz y José Pedro

A mis hermanas, Paz y Ana

A mi novia, Begoña

AGRADECIMIENTOS

A mis padres que con sus consejos y apoyo constante me han ayudado a formarme como persona y académicamente. Gracias por todos los esfuerzos que habéis tenido que realizar para que pudiera hacer lo que me gusta en todo momento. Gracias por guiarme por el camino correcto y corregirme cuando perdía su senda y me desviaba por rutas erróneas. Gracias en definitiva por todo.

A mis hermanas por su cariño en todo momento y por aguantarme sobre todo en los meses de vacaciones en los que les daba la tabarra más de la cuenta.

A mi novia por estar ahí a lo largo de todos estos años, por hacerme compañía en todo momento y soportarme en las largas épocas de exámenes. Por animarme cuando más lo necesitaba y creer siempre en mí ayudándome a superar los momentos más difíciles.

A mi tutor Agapito por aceptar dirigirme el proyecto, por las horas dedicadas a él y por todos los consejos proporcionados para poder llevarlo a cabo.

A todos los miembros de mi familia que me han apoyado a lo largo de todos estos años lejos de casa. A mis abuelos Chelo, Maruja, Pepe y Ángel, a mis tías, tíos, primas y primos por ayudarme siempre que lo he necesitado.

A mis compañeros de universidad con los que tantos momentos de angustia he compartido a la hora de entregar cada una de las prácticas y trabajos llevados a cabo a lo largo de la carrera.

A mis amigos por hacerme pasar tantos buenos ratos y estar ahí siempre que les he necesitado.

A Quico, integrante del grupo Los Pelayos por haberme aconsejado, guiado y por proporcionarme parte de los datos necesarios para llevar a cabo este proyecto.

En definitiva gracias a todos los que me han ayudado a lo largo de estos años universitarios que tan deprisa han pasado.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1: INTRODUCIÓN

La ingeniería informática desde el aspecto lógico y formal está fundamentada en la teoría de autómatas, los lenguajes formales, la teoría de la información, el diseño de algoritmos, el reconocimiento de patrones, la inteligencia artificial (IA) y la ingeniería del conocimiento. Este proyecto está centrado en una pequeña parte de la IA.

Farid Fleifel Tapia describe a la IA como la rama de la ciencia de la computación que estudia la resolución de problemas no algorítmicos mediante el uso de cualquier técnica de computación disponible, sin tener en cuenta la forma de razonamiento subyacente a los métodos que se apliquen para lograr esa resolución (Tapia).

La IA tiene numerosas aplicaciones entre otras están las industriales, médicas, mundos virtuales, procesamiento de lenguaje natural, robótica, videojuegos y un largo etcétera. Este trabajo únicamente se centrará en la minería de datos (del inglés *data mining*) con técnicas de IA.

La minería de datos (DT) consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. En otras palabras, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

En este caso, para la realización de este trabajo se utilizó la metodología CRISP-DM (CRoss Industry Standard Process, 1996) que se presentará en apartados posteriores.

Una vez definida la rama de la ingeniería informática en la que se enmarcó este proyecto a continuación, se resume el campo a tratar. En este proyecto se analizó la viabilidad de aplicar un proceso de minería de datos utilizando técnicas de inteligencia artificial para la predicción de los resultados de eventos deportivos. Se contemplaron varios tipos de deportes pero debido a la gran cantidad de datos disponible hubo que acotar el dominio, ya que, no fue posible analizar los datos de todos los deportes contemplados, debido a que, el tamaño del estudio era superior a lo requerido en un proyecto fin de carrera (PFC). En este caso, el proyecto se centró únicamente en el

deporte del tenis a nivel masculino e individual, por ser dentro de este deporte la modalidad de la que a priori más datos se disponían, ya sea personales de los tenistas, estadísticas de los partidos o movimientos de apuestas. La elección de este deporte fue debida a que en su apuesta principal (ganador de un partido) sólo admitía dos posibles resultados lo que simplificó la predicción.

1.1 Objetivos del proyecto

El estudio expuesto en este documento lleva a cabo un proceso de descubrimiento de conocimiento mediante técnicas de inteligencia artificial en el dominio de las apuestas en eventos deportivos. En concreto el estudio se centra en el tenis masculino e individual. Para realizar dicho estudio se siguió la metodología CRISP-DM y se tuvieron como principales objetivos los siguientes:

- Aplicar distintas técnicas de minería de datos, para analizar el gran volumen de datos disponible de tenistas, sus partidos y los movimientos que estos generan en el mercado de las apuestas deportivas.
- Conocer qué factores son más determinantes en un partido de tenis y en su resultado final.
- Conocer el movimiento de las apuestas de un partido de tenis y analizar si este conocimiento puede ser útil para una posible inversión en las apuestas.
- Diseñar un sistema automático mediante el cual se consiga maximizar la ganancia de una posible inversión en apuestas deportivas gracias al conocimiento extraído en la predicción de partidos de tenis y del movimiento de las apuestas.

1.2 Estructura y contenido del documento

En el capítulo dos se describirá el estado del arte exponiendo las técnicas y metodología empleadas para llevar a cabo el proyecto. En este caso se presentará una breve introducción a la minería de datos y dónde está encuadrada la metodología CRISP-DM dentro de la misma. A continuación, se describirán las distintas fases que establece CRISP-DM. También se repasarán los tipos de problemas tratados por la minería de datos y por último se describirá la historia del dominio de este estudio.

Los siguientes capítulos en los que se estructura el documento son fruto del seguimiento de la metodología CRISP-DM que fue la elegida para llevar a cabo el proyecto.

El capítulo tres se enfocará en la comprensión de los objetivos del proyecto y exigencias desde una perspectiva de negocio, para definir un problema de minería de datos y elaborar un plan preliminar diseñado para alcanzar dichos objetivos.

En el capítulo cuatro se describirá la fase de comprensión de los datos que comenzó con la colección de datos inicial y continuó con las actividades que permitieron la familiarización con los datos.

El quinto capítulo cubrirá todas las actividades necesarias para conformar el conjunto de datos final de los datos en brutos iniciales. Las tareas de preparación de los datos fueron realizadas muchas veces y sin un orden prescripto, e incluyeron la selección de registros y atributos, así como la transformación y limpieza de la información.

En el capítulo seis varias técnicas de modelado fueron seleccionadas y aplicadas. Fue a menudo necesario, de acuerdo a los algoritmos y técnicas seleccionados, volver a la fase de preparación de los datos.

En el séptimo capítulo, se evaluarán los modelos construidos. Un objetivo clave fue determinar si había alguna cuestión importante de negocio que no hubiese sido considerada suficientemente. Al final de este capítulo, se tomó una decisión sobre el uso de los resultados de minería de datos, detallando si los modelos construidos serán válidos para la vida real.

El capítulo octavo expondrá las conclusiones más importantes y las propuestas de trabajos futuros que podrían ampliar lo estudiado en el presente proyecto.

Finalmente se presentarán tanto un glosario de términos y acrónimos como la bibliografía consultada para la elaboración del proyecto.

Por último se incluirán algunos anexos útiles para la lectura de este texto.

Capítulo 2:
ESTADO DEL ARTE

Hoy en día existen ingentes volúmenes de información estructurada, sobre actividades de todo tipo. En esta información acumulada se pueden rastrear asociaciones y patrones que es imposible observar a simple vista, por ello es necesario aplicar distintas técnicas con plataformas informáticas para poder identificar las posibles relaciones. El análisis manual de los datos es caro, ya que, consume muchos recursos en especialistas que no por ser especialistas llevan a cabo el análisis en poco tiempo debido a la necesidad de formular hipótesis, probarlas, ajustarlas, volverlas a probar y así sucesivamente. Además, cada vez los análisis conllevan un número mayor de parámetros y en consecuencia de relaciones entre ellos. Es por todo esto que entra en juego la minería de datos.

2.1 Minería de datos y CRISP-DM

El boom de la informática en la última década ha traído consigo un aumento del volumen y variedad de la información que se encuentra informatizada en bases de datos digitales. Gran parte de esta información es histórica, es decir, representa transacciones o situaciones que se han producido en el pasado. Aparte de su función de “memoria de la organización”, la información histórica es útil para predecir la información futura.

Es común que empresas, instituciones, organizaciones, particulares, etc tomen decisiones en base a información obtenida de experiencias pasadas, esta información puede venir de fuentes muy distintas. Hoy en día, muchas decisiones se basan en el análisis de datos, cuyo volumen desborda la capacidad humana, es por ello que el área de extracción de conocimiento semiautomático ha adquirido, recientemente, una gran importancia.

Bajo el nombre de minería de datos se engloba todo un conjunto de técnicas encaminadas a la extracción de conocimiento procesable, implícito en las bases de datos. Está fuertemente ligado con la supervisión de procesos industriales ya que resulta muy útil para aprovechar los datos almacenados en las bases de datos. Las bases de la minería de datos se encuentran en la inteligencia artificial y en el análisis estadístico. Mediante los modelos extraídos utilizando técnicas de minería de datos se aborda la solución a problemas de predicción, clasificación y segmentación (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

Dentro de la minería de datos existen numerosas metodologías entre las que se encuentra CRISP-DM, metodología seguida en este proyecto.

2.1.1 CRISP-DM

Durante 1996 el interés en la minería de datos iba creciendo pero se trataba de una industria joven e inmadura lo que provocaba que los acercamientos a este tipo de proyectos fueran dubitativos. A finales de este año tres líderes de la industria: DaimlerBenz, SPSS (entonces ISL) y NCR formaron un consorcio, inventaron un acrónimo CRISP-DM (CRoss-Industry Standard Process for Data Mining) y comenzaron a proponer ideas. Actualmente existen alrededor de 200 miembros del CRISP-DM Special Interest Group (SIG), incluidos proveedores de DM, consultores y usuarios finales (CRoss Industry Standard Process, 1996).

Distribución jerárquica

La metodología CRISP-DM está descrita en términos de un proceso jerárquico consistente en un conjunto de tareas descritas en cuatro niveles de abstracción (de lo general a lo específico): fase, tarea genérica, tarea especializada, e instancia de proceso (ver Figura 1).

En el nivel superior, el proceso de minería de datos está organizado en un número de fases; cada fase está formada por varias tareas genéricas que forman el segundo nivel. Este segundo nivel se denomina genérico porque está destinado a ser bastante general para cubrir todas las situaciones posibles de la minería de datos. Las tareas genéricas están destinadas a ser tan completas y estables como sea posible. Se entiende en este caso por completo que cubre tanto al proceso entero de minería de datos como a todas las aplicaciones de minería de datos posibles. Estable significa que el modelo debería ser válido para acontecimientos normales y también para desarrollos imprevistos como nuevas técnicas de modelado.

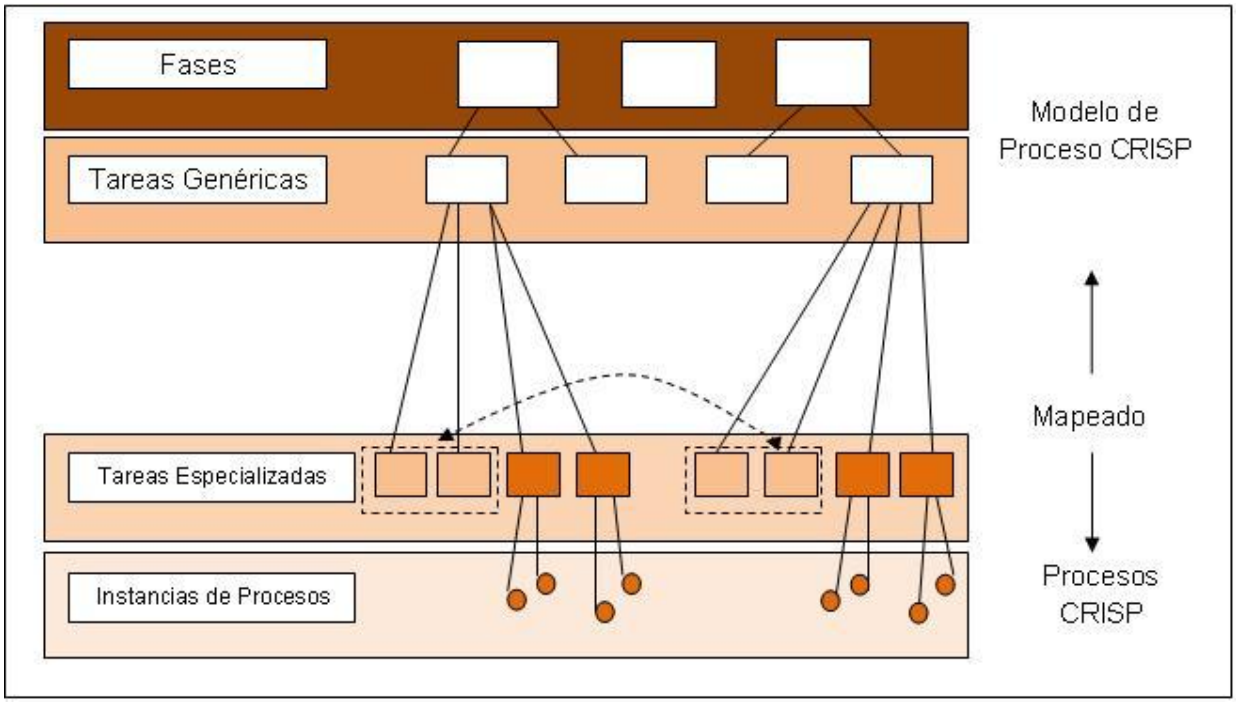


Figura 1: Desglose de la metodología CRISP-DM en 4 niveles.

El tercer nivel, el nivel de tareas especializadas, sirve para describir como deberían ser realizadas, en ciertas situaciones específicas, las acciones en las tareas genéricas. Por ejemplo, en el segundo nivel podría haber una tarea genérica llamada limpieza de datos, el tercer nivel describe cómo esta tarea se diferencia en situaciones distintas, distinguiendo por ejemplo la limpieza de valores numéricos de la limpieza de valores categóricos, o si el tipo de problema es de agrupamiento o predicción.

La descripción de fases y tareas como pasos discretos realizados en un orden específico representa una secuencia idealizada de eventos.

En la práctica, muchas de las tareas pueden ser realizadas en un orden diferente siendo a menudo es necesario volver a hacer tareas anteriores repitiendo ciertas acciones. El modelo de proceso no intenta capturar todas estas posibles rutas del proceso de la minería de datos porque esto requeriría un modelo de proceso demasiado complejo.

El cuarto nivel, las instancias de procesos, es un registro de las acciones, las decisiones y los resultados de la minería de datos real.

Una instancia de proceso se organiza de acuerdo a las tareas definidas en los niveles superiores, pero representa lo que en realidad ocurre en un caso concreto, en lugar de lo que sucede en general.

Modelo de referencia y guía de usuario

Horizontalmente, en la metodología CRISP-DM se distingue entre el modelo de referencia y la guía de usuario. El modelo de referencia presenta una descripción

rápida de las fases, tareas, y sus salidas, describiendo qué hacer en el proyecto de minería de datos. La guía de usuario da consejos más detallados e insinuaciones para cada fase y cada tarea dentro de una fase, también indica cómo realizar un proyecto de minería de datos siguiendo la metodología. A continuación, se describirá únicamente el modelo de referencia, sin embargo, a lo largo de la realización del proyecto la guía de usuario también fue consultada.

2.1.2 Paso de modelos genéricos a especializados

El contexto de minería de datos traza un mapa entre el nivel genérico y el especializado en CRISP-DM. Actualmente, se distingue entre cuatro dimensiones diferentes de contextos de minería de datos:

- El **dominio de aplicación** es el área específica en la que el proyecto de minería de datos toma lugar.
- Los **tipos de problemas de minería de datos** describen la(s) clase(s) específica(s) de objetivo(s) con el que el proyecto de minería de datos trata.
- El **aspecto técnico** cubre cuestiones específicas en la minería de datos que describen las distintas dificultades (técnicas) que por lo general ocurren durante el proyecto de minería de datos.
- La **herramienta y dimensión técnica** específicas que se aplican como herramienta y/o técnica de minería de datos durante el proceso de minería de datos.

A continuación, en la Tabla 1 se muestran ejemplos de las cuatro dimensiones de contextos de minería de datos.

	Contexto de Minería de Datos			
Dimensión	Dominio de aplicación	Tipo de problema de Minería de Datos	Aspectos técnicos	Herramientas y técnicas
Ejemplos	Modelo de respuesta	Descripción y resumen	Valores Perdidos	Clementine
	Predicción de rotación	Segmentación	Valores extremos	MineSet
	…	Descripción de conceptos	…	Árbol de decisión
		Clasificación		…

		Predicción		
		Análisis de dependencias		

Tabla 1: Dimensiones de contextos de minería de datos y ejemplos.

Un contexto específico de minería de datos es un valor concreto para una o más de estas dimensiones.

En CRISP-DM se distinguen dos tipos de correspondencia entre en el nivel genérico y el especializado.

- **Correspondencia para el presente**: Si sólo se aplica el modelo de proceso genérico para realizar un proyecto de minería de datos simple, intentando pasar de las tareas genéricas y sus descripciones al proyecto específico según las necesidades, se habla de una asignación para (probablemente) un sólo uso.
- **Correspondencia para el futuro**: Si sistemáticamente se especializa el modelo de proceso genérico según un contexto predefinido, se habla explícitamente de la sobre escritura de un modelo de proceso especializado en términos de CRISP-DM.

Cualquiera de los tipos de correspondencia es apropiado, para según qué objetivos, dependiendo del contexto de la minería de datos específico y de las necesidades de la organización.

La estrategia básica para pasar del modelo de proceso genérico al nivel especializado es la misma para ambos tipos de correspondencia:

- Analizar el contexto específico.
- Quitar cualquier detalle no aplicable al contexto.
- Agregar cualquier detalle específico al contexto.
- Especializar (o instanciar) el contenido genérico según las características concretas del contexto.
- Renombrar el contenido genérico si es posible, para proporcionar significados más explícitos en el contexto y de esta forma obtener mayor claridad.

2.1.3 Modelo de referencia de CRISP-DM

El modelo de proceso corriente para la minería de datos proporciona una descripción del ciclo de vida del proyecto de minería de datos. Éste contiene las fases

de un proyecto, sus respectivas tareas y las relaciones entre estas tareas. En este nivel de descripción, no es posible identificar todas las relaciones pero, si cabe mencionar que podrían existir relaciones entre cualquier tarea de minería de datos según los objetivos, los antecedentes, el interés del usuario y, lo más importante, los datos.

El ciclo de vida del proyecto de minería de datos cuenta con seis fases sin una secuencia rígida como se puede observar en la Figura 2.

El movimiento hacia adelante y hacia atrás entre las distintas fases es siempre necesario. El resultado de cada fase determina qué fase, o tarea particular de una fase, tienen que ser realizadas después. Las flechas indican las más importantes y frecuentes dependencias entre fases.

El círculo externo de la Figura 2 simboliza la naturaleza cíclica de la propia minería de datos. La minería de datos no se termina una vez que la solución es desplegada. Las informaciones ocultas durante el proceso y la solución desplegada pueden provocar nuevas preguntas. Los procesos de minería de datos sucesivos se beneficiarán de las experiencias previas.

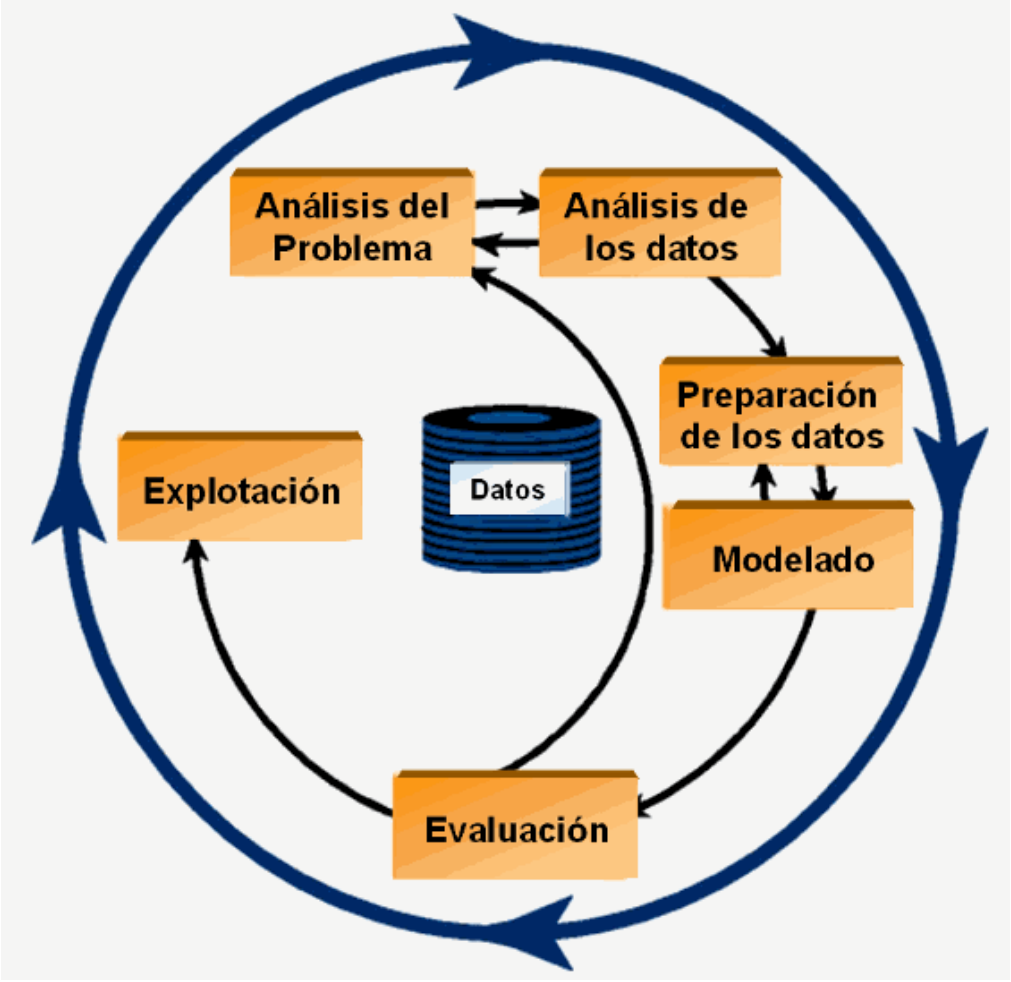


Figura 2: Fases del modelo de referencia CRISP-DM.

La metodología propone un orden lógico a sus fases, aunque permite retrocesos entre varias de ellas, pues frecuentemente, a lo largo del desarrollo de un proyecto, es necesario volver atrás en numerosas ocasiones para re-analizar los resultados obtenidos.

Además, el proyecto se torna cíclico, pues, éste no se termina una vez que la solución es desplegada, ya que las informaciones obtenidas pueden provocar nuevas preguntas. Los procesos de minería de datos posteriores se beneficiarán de las experiencias previas.

A continuación, se resumen las distintas fases que componen CRISP-DM:

Análisis del problema/Comprensión del Negocio

Esta fase inicial se enfoca en la comprensión de los objetivos y exigencias del proyecto, desde una perspectiva de negocio, para definir un problema de minería de datos y elaborar un plan preliminar diseñado para alcanzar dichos objetivos.

- **Determinación de los objetivos de negocio**: El primer objetivo del analista de datos para un contexto es entender, desde una perspectiva de negocio, lo que el cliente realmente quiere lograr. Se describen los criterios de éxito para un resultado acertado o útil para el proyecto desde el punto de vista del negocio.

- **Evaluación de la situación**: Se enuncian los recursos disponibles para el proyecto (personal, datos, recursos computacionales, otros). Se realiza un cronograma del proceso, se enumeran las presunciones, restricciones y disponibilidad de recursos. Se listan los riesgos que podrían retrasar el proyecto y los planes de contingencia correspondientes. Se realiza un análisis de costo-beneficio para el proyecto, tan específico como sea posible.

- **Determinación de los objetivos de la minería de datos**: Esta tarea tiene como objetivo representar los objetivos del negocio en términos de las metas del proyecto de minería de datos. Se definen los criterios de éxito de un resultado exitoso para el proyecto en términos técnicos, por ejemplo, un cierto nivel de predicción. Además, también puede expresarse en términos subjetivos, y en este caso, deben ser identificadas las personas que hacen el juicio.

- **Elaboración del plan del proyecto**: Se describe el plan para alcanzar los objetivos de minería de datos y con ello los del negocio; dicho plan debe especificar los pasos durante el resto del proyecto, incluyendo la selección inicial de herramientas y técnicas, y una lista de las etapas a ser ejecutadas, junto con su duración, recursos requeridos, entradas, salidas y dependencias.

Comprensión de los datos

La segunda, fase de comprensión de los datos, comprende la recolección inicial de datos, con el objetivo de establecer un primer contacto con el problema, familiarizarse con ellos, identificar su calidad y establecer las relaciones más evidentes que permitan definir las primeras hipótesis. Esta fase junto a las próximas dos fases, son las que demandan el mayor esfuerzo y tiempo en un proyecto de minería de datos.

Las principales tareas a desarrollar en esta fase del proceso son:

- **Recolección inicial de los datos:** Se confecciona una lista del conjunto de datos obtenidos, sus localizaciones y los métodos usados para obtenerlos.
- **Descripción de los datos iniciales**: Se describen los datos que han sido adquiridos, incluyendo su formato y cantidad; además, se evalúa si satisfacen las exigencias previstas.
- **Exploración de los datos:** Esta tarea está dirigida a responder interrogantes de minería de datos usando visualización y técnicas de reporte. De ser apropiado pueden ser incluidos gráficos para indicar las características de los datos, de donde se desprenden las conclusiones o hipótesis iniciales del proyecto
- **Verificación de la calidad de los datos:** Se examina la calidad de los datos en relación a si están completos, si son correctos, si contienen errores y qué tan comunes son estos, si existen valores omitidos, etc.

Preparación de los Datos

Esta fase cubre todas las actividades necesarias para conformar el conjunto de datos final (los datos que serán utilizados por las herramientas de modelado) a partir de los datos en bruto iniciales. Las tareas de preparación de datos probablemente van a ser realizadas muchas veces y sin un orden preestablecido, e incluyen la selección de registros y atributos, así como el proceso de transformación y limpieza.

- **Selección de los datos**: Se decide qué datos serán excluidos y cuáles usados para el análisis, de acuerdo a su importancia respecto a los objetivos de la minería de datos, su calidad y las restricciones técnicas. Cubre la selección de atributos (columnas) así como la selección de registros (filas).
- **Limpieza de los datos**: Se debe elevar la calidad de los datos al nivel requerido por las técnicas de análisis seleccionadas. Esto puede implicar la selección de subconjuntos de datos limpios, la inserción de datos por defecto adecuados, o técnicas más ambiciosas tales como la estimación de datos ausentes mediante modelado.

- **Construcción de los datos**: Esta tarea incluye la construcción de operaciones de preparación de datos tales como la producción de atributos derivados, el ingreso de nuevos registros o la transformación de valores para atributos existentes.

- **Integración de los datos**: Son los métodos por el cual la información es combinada de múltiples tablas o registros para crear nuevos registros o valores.

- **Formato de los datos:** Se realizan modificaciones principalmente sintácticas a los datos que no cambian su significado, pero que si pueden ser requeridas por la herramienta de modelado.

Modelado

En esta fase, varias técnicas de modelado son seleccionadas y aplicadas, y sus parámetros son ajustados. Es a menudo necesario, de acuerdo a los algoritmos y técnicas seleccionadas, volver a la fase de preparación de los datos.

- **Selección de las técnicas de modelado:**

  - Técnica de modelado: Como primer paso durante el modelado, se debe seleccionar la técnica de modelado que será usada. Si son aplicadas múltiples técnicas, se realiza esta tarea de forma individual para cada una de ellas.

  - Suposiciones del modelado: Se registra cualquier presunción de la técnica de modelado seleccionada, que pueden ser, por ejemplo, que todos los atributos tengan distribuciones uniformes, que el atributo a predecir deba ser simbólico, etc.

- **Generación del diseño del experimento**: se describe el plan intencionado para el entrenamiento, la prueba y la evaluación de los modelos. Un componente primario del plan determina como dividir un conjunto de datos disponible en datos de entrenamiento y datos de validación.

- **Construcción y descripción de los modelos:**

  - Escenario de parámetros: Se listan los parámetros y los valores escogidos para los mismos, así como el razonamiento llevado a cabo para elegirlos.

  - Modelos: Se listan los modelos reales producidos por la herramienta de modelado, no un informe.

  - Descripción de los modelos: Se describen los modelos obtenidos, informándose su interpretación y documentándose cualquier dificultad encontrada con sus significados.

- **Evaluación del modelo**: Se resumen los resultados de esta tarea, listando las calidades de los modelos generados y comparando unos con otros.

Evaluación

En esta etapa, se evalúan los modelos construidos, revisando cada uno de los pasos ejecutados para crearlos, a fin de comprobar si cumplen correctamente con los objetivos del negocio. Un objetivo clave es determinar si hay alguna cuestión importante del negocio que no ha sido considerada suficientemente. Al final de esta fase, se toma una decisión sobre el uso de los resultados obtenidos en el proceso de minería de datos.

- **Evaluación de los resultados:**
  - Valoración de la minería respecto al negocio: Se resumen los resultados de minería de datos en términos de criterios de éxito del negocio.
  - Aprobación de los modelos: Después de la valoración de los modelos, se toma una decisión al respecto.
- **Revisión del proceso:** Se califica el proceso entero de minería de datos con el objetivo de identificar elementos que pudieran ser mejorados.
- **Determinar los próximos pasos:** Si se ha determinado que las fases, hasta este momento, han generado resultados satisfactorios, podría pasarse a la siguiente fase o, en caso contrario, podría decidirse realizar otra iteración desde la fase de preparación de los datos o de modelado. Podría ser incluso que en esta fase se decida partir desde cero con un nuevo proyecto de minería de datos.

Despliegue

La fase de despliegue puede ser tan simple como la generación de un informe o tan compleja como la repetición del proceso de minería a través de la organización. En muchos casos, es el cliente y no el analista de datos, quien lleva a cabo la fase de despliegue, sin embargo, resulta conveniente la participación de ambos para comprender rápidamente qué acciones ejecutar a fin de emplear los modelos obtenidos.

- **Planificación del despliegue:** De acuerdo al desarrollo de los resultados de la minería en el negocio, se determina una estrategia para su despliegue, donde se incluyen los pasos necesarios y cómo realizarlos.
- **Planificación de la monitorización y el mantenimiento:** Se resume la estrategia de supervisión y mantenimiento, incluyendo los pasos

necesarios y cómo realizarlos, a fin de evitar largos periodos innecesarios de uso incorrecto de los resultados de minería de datos.

- **Generación del informe final:** Se redacta un informe escrito final del compromiso de la minería de datos, lo que incluye todo el desarrollo anterior, el resumen y la organización de los resultados. A menudo se realiza una reunión al finalizar en la que los resultados son presentados verbalmente
- **Revisión del proyecto:** De igual modo se resumen las experiencias importantes ganadas durante el proyecto.

2.1.4 Tipos de problemas de la minería de datos

Por lo general, los proyectos de minería de datos implican una combinación de diferentes tipos de problemas, que juntos solucionan el problema de negocio.

Descripción de datos y resumen

La descripción y el resumen de datos apuntan a la especificación concisa de las características de los datos, típicamente en forma elemental y agregada. Esto da al usuario una exposición de la estructura de los datos. A veces, una descripción y resumen de los datos puede ser el objetivo de un proyecto de minería de datos.

En casi todos los proyectos de minería de datos, sin embargo, la descripción y resumen de los datos son un objetivo subordinado en el proceso, típicamente en sus etapas tempranas. Al principio de un proceso de minería de datos, el usuario a menudo no conoce, ni el objetivo preciso del análisis, ni la naturaleza exacta de los datos. La exploración inicial del análisis de datos puede ayudar a los usuarios a entender la naturaleza de los datos y formar hipótesis potenciales de la información oculta. La estadística descriptiva simple y las técnicas de visualización proporcionan las primeras ideas sobre los datos.

La descripción y el resumen de datos típicamente ocurren en combinación con otros tipos de problemas de minería de datos. Es aconsejable llevar a cabo una descripción y resumen de datos antes de que cualquier otro tipo de problema de minería de datos sea especificado. El resumen también juega un papel importante en la presentación de los resultados finales.

Segmentación

La segmentación apunta a la separación de los datos en subgrupos o clases significativas e interesantes. Todos los miembros de un subgrupo comparten características comunes.

La segmentación puede ser realizada a mano o semiautomáticamente. El analista puede suponer ciertos subgrupos como relevantes para la pregunta de negocio, basada sobre un conocimiento previo o sobre el resultado de la descripción y el resumen de datos. En adición, hay también técnicas automáticas de agrupamiento (del inglés *clustering*) que pueden descubrir estructuras antes insospechadas y ocultas en datos que permiten la segmentación.

La segmentación a veces puede ser un objetivo de la minería de datos. Entonces la detección de segmentos sería el objetivo principal de un proyecto de minería de datos.

Muy a menudo, sin embargo, la segmentación es un paso hacia la solución de otros tipos de problema. Entonces, el objetivo es el de guardar o mantener el tamaño de los datos manejables o encontrar los subconjuntos de datos homogéneos que son más fáciles para analizar.

Entre las técnicas apropiadas para la segmentación se puede señalar:

- Técnicas de agrupamiento.
- Redes Neuronales.
- Visualización.

Descripciones de concepto

La descripción de concepto apunta a una descripción comprensible de conceptos o clases. El objetivo de las descripciones de concepto no es completar el desarrollo de modelos con predicción de exactitud alta, sino obtener ideas.

Una descripción de concepto tiene una conexión cercana tanto a la segmentación como a la clasificación. La segmentación puede conducir a una enumeración de objetos que pertenecen a un concepto o clase sin proporcionar cualquier descripción comprensible. Típicamente la segmentación es llevada a cabo antes de que la descripción de concepto sea realizada.

Las descripciones de concepto también pueden ser usadas con objetivos de clasificación. Por otra parte, algunas técnicas de clasificación producen modelos de clasificación comprensibles, que pueden entonces ser consideradas descripciones de concepto. La distinción importante es que la clasificación apunta a ser completa en algún sentido. El modelo de clasificación tiene que aplicarse a todos los casos en la población seleccionada.

Por otro lado, las descripciones de concepto no tienen que ser completas. Es suficiente si describen las partes importantes de los conceptos o clases. Técnicas apropiadas de descripción son:

- Métodos de inducción de reglas.
- Agrupamiento conceptual.

Clasificación

La clasificación asume que hay un conjunto de objetos caracterizados por algún atributo o rasgo que pertenece a diferentes clases. La etiqueta de clase es un valor (simbólico) discreto y es conocido para cada objeto. El objetivo es construir los modelos de clasificación (a veces llamados clasificadores), que asignan la etiqueta de clase correcta a objetos no vistos antes y sin etiquetas. Los modelos de clasificación sobre todo son usados para el modelado predictivo.

Las etiquetas de clase pueden ser presentadas antes de la segmentación, o como consecuencia de ella. La clasificación es uno de los tipos de problemas más importantes de minería de datos que están presentes en una amplia gama de aplicaciones. Muchos problemas de minería de datos pueden ser transformados en problemas de clasificación

La clasificación tiene conexiones con casi todos los otros tipos de problemas. Los problemas de regresión pueden ser transformados en problemas de clasificación por discretización de etiquetas de clase continuas, porque las técnicas de discretización permiten transformar rangos continuos en intervalos discretos. Estos intervalos discretos, más que los valores numéricos exactos, son usados como etiquetas de clase, lo que lleva a un problema de clasificación. Algunas técnicas de clasificación producen una clase comprensible o descripciones de concepto. Hay también una conexión al análisis de dependencias porque los modelos de clasificación típicamente usan (explotan) y aclaran las dependencias entre atributos.

La segmentación puede también proporcionar las etiquetas de clase o restringir el conjunto de datos para que se puedan construir buenos modelos de clasificación. Un modelo de clasificación también puede ser usado para identificar desviaciones y otros problemas con los datos.

Entre las técnicas de clasificación se encuentran:

- Análisis discriminante.
- Métodos de inducción de reglas.
- Árboles de Decisión.
- Redes neuronales.
- Vecino más cercano.
- Razonamiento basado en casos.
- Algoritmos genéticos.

Regresión

Otro tipo de problema importante que ocurre en una amplia gama de usos es la regresión. La regresión es muy similar a la clasificación.

La única diferencia es que en la regresión el atributo objetivo (la clase) no es un atributo cualitativo discreto, sino que es uno continuo.

El objetivo de la regresión está en encontrar el valor numérico del atributo objetivo para objetos no vistos. Si la regresión trata con datos de series temporales, entonces a menudo se llama estimación.

Algunas técnicas de regresión destacadas son:

- Análisis de regresión.
- Árboles de regresión.
- Redes neuronales.
- El vecino más cercano.
- Métodos de la Caja-Jenkins.
- Algoritmos genéticos.

Análisis de dependencias

El análisis de dependencias consiste en encontrar un modelo que describa dependencias significativas (o asociaciones) entre datos o elementos. Las dependencias pueden ser usadas para predecir el valor de un elemento dada la información de otros. Las dependencias pueden ser estrictas o probabilísticas.

Las asociaciones son un caso especial de dependencias, que recientemente se han hecho muy populares. Las asociaciones describen las afinidades entre elementos. Los algoritmos para detectar asociaciones son muy rápidos y producen muchas asociaciones. Seleccionar el más interesante es un desafío.

El análisis de dependencias tiene conexiones cercanas a la regresión y a la clasificación, ya que las dependencias implícitamente son usadas para la formulación de modelos predictivos. Hay también una conexión con las descripciones de concepto, que a menudo destacan dependencias. El modelo secuencial es una clase especial de dependencias en las que el orden de acontecimientos es considerado.

Algunas técnicas de análisis de dependencias destacadas son:

- Análisis de correlación.
- Análisis de regresión.
- Reglas de asociación.
- Redes bayesianas.
- Programación de lógica inductiva.
- Técnicas de visualización

2.2 Historia de las apuestas

Muchos son los esfuerzos realizados por personas a nivel individual o en conjunto, en las denominadas peñas, para tratar de acertar la conocida quiniela futbolística, asentada en nuestro país desde el año 1946. Mientras aquí, hasta hace pocos años, sólo había dos medios para apostar legalmente a eventos deportivos (la quiniela y la hípica, este último apostando en directo en el hipódromo), en otros países ya existían las casas de apuestas.

Los primeros colonizadores ingleses llegaron a Estados Unidos con el juego en sus venas. Después de todo, sus padres y abuelos habían apostado en todo tipo de eventos además de en eventos deportivos. No solamente buscaban una ganancia financiera, sino una forma de ocio y entretenimiento.

En aquellos años, las personas apostaban por el resultado de carreras de caballos, peleas a puñetazos o peleas de gallos. No pasó mucho tiempo hasta que se empezó a apostar cada vez más a eventos deportivos.

Las carreras de caballos disfrutaron de su más alta popularidad durante los siglos XIX y XX. Aunque la actividad fue disfrutada principalmente por la clase alta, después de la Guerra Civil, los hipódromos comenzaron a surgir por todo el este de Estados Unidos con cada vez más entusiastas del juego. A partir de ahí, apostar a los caballos en los hipódromos era posible para todo el mundo, independientemente de cuál fuera su situación económica.

En el siglo XIX, el negocio de las carreras de caballos en América, llegó a su pico con más de 300 pistas de carreras operando y admitiendo apuestas. Además, empezaron a surgir las ahora conocidas como casas de apuestas, que se encontraban fuera de los hipódromos, éstas estaban conectadas con los hipódromos a través de cables telegráficos. No importaba dónde estuvieran localizadas las casas de apuestas, los apostadores ya no tenían que estar físicamente en el hipódromo para efectuar sus apuestas.

A finales del siglo XIX, el béisbol profesional aumentó su popularidad y junto con él, las apuestas sobre los resultados de los partidos. Los boletos de apuestas se encontraban por todo el este del país pudiendo apostar a gran variedad de eventos.

Antes de 1958, el sector de las apuestas de las carreras de caballos, estuvo restringido a lo que comúnmente se conoce como los pequeños *turf clubs*. Con los deportes televisados las apuestas proliferando durante las décadas del 60 y 70.

En 1947 el gobierno de Estados Unidos eliminó un 10% del impuesto de las apuestas deportivas, alterando significativamente el curso de su historia. Como resultado directo, los casinos importantes comenzaron inmediatamente a abrir lugares de sellado de apuestas como una forma de aumentar sus ingresos generales.

Con el boom de internet no ha hecho falta esperar mucho tiempo para que las apuestas deportivas hayan llegado a alturas sin precedentes. Hoy, apostar en

competiciones deportivas, está más de moda que nunca, tanto en línea como en locales comerciales.

Según cita un reportaje de la ESPN (*Entertainment & Sports Programming Network*), el negocio de las apuestas deportivas en línea ganó aproximadamente 63 billones de dólares en el 2003. Además también estimaba que un 25% de los ciudadanos estadounidenses apostaban en eventos deportivos por lo menos una vez al año siendo el 15% los que apostaban en eventos deportivos regularmente. Hoy en día es muy probable que estos resultados hayan aumentado considerablemente.

El sector de las apuestas por Internet en España habría cerrado 2008 con una facturación superior a los 200 millones de euros, frente a los 170 millones de euros del ejercicio anterior, según las estimaciones realizadas por la Asociación Española de Apostadores por Internet (Aedapi). La asociación también estimó que esta tendencia creciente se mantendrá en los próximos años llegando a facturar hasta 450 millones de euros en 2012.

2.2.1 Casas de intercambio de apuestas

El fenómeno del intercambio de apuestas deportivas es reciente. Una casa de intercambio de apuestas deportivas es un sitio web *p2p* (de persona a persona), que actúa como un intermediario entre partes en la colocación de apuestas. El concepto es similar al de la bolsa de valores, pero en este caso el bien que es comercializado no es una acción de una empresa, sino una apuesta.

El funcionamiento del intercambio de apuestas deportivas es sencillo. Del mismo modo que los sitios de subastas juntan compradores y vendedores a través de Internet, los sitios de intercambio de apuestas juntan clientes que quieren apostar en determinados eventos. Por lo tanto, el apostante no cruza su apuesta con la casa de apuestas, sino que la cruza con otro apostante, siendo la casa de intercambio de apuestas un intermediario entre ambos apostantes.

La casa de intercambio de apuestas deportivas asume un papel pasivo en la elección de las cuotas y las cantidades que son apostadas, ya que son los propios apostantes quienes las deciden de mutuo acuerdo. Las casas de intercambio generan sus ingresos al cobrar una comisión la cual es calculada como una cuota fija por transacción o más comúnmente por porcentajes de las ganancias netas de cada cliente.

La casa de intercambio de apuestas y la casa de apuestas ofrecen dos alternativas de apostar diferentes, pero ambas, tienen sus pros y sus contras.

En una casa de intercambio de apuestas las ventajas son:

- Mejores cuotas: Como uno apuesta contra otros apostantes, no hay necesidad de pagar los enormes márgenes de las casas de apuestas

tradicionales. En promedio las cuotas a favor son un 20% mejores que las de las casas de apuestas tradicionales.

- No hay limitaciones: En el intercambio de apuestas los apostantes no son penalizados por ser exitosos. Sin embargo, las casas de apuestas tradicionales limitan la cantidad a ser apostada en cada evento y las cantidades máximas de ganancias de un usuario para un periodo de tiempo determinado.

- Variedad: Además de poder apostar en forma tradicional a favor de una selección (*back*), en el caso del intercambio de apuestas es posible apostar en contra de una selección (*lay*). ¿Qué significa esto? Que se puede apostar a que, por ejemplo, el Zaragoza va a ser vencido en un partido, por lo cual se obtendrían ganancias tanto si perdiera como si empatara el encuentro.

- Control: No es necesario tomar las apuestas disponibles en el momento. Si se quiere pedir mejores cuotas, simplemente se puede dejar la oferta en el sistema y esperar a que otros apostantes la igualen. Pero hay que ser realista, ya que debe haber alguien preparado para apostar en contra de la apuesta con esas cuotas, y viceversa.

Los inconvenientes que se pueden encontrar en una casa de intercambio de apuestas son:

- Publicación de las cuotas: las casas de apuestas fijan la cuota de salida de antemano abriendo el mercado rápidamente, en cambio, en la casa de intercambio son los propios usuarios los que definen la cuota, por lo tanto, tardan más en abrirse.

- Liquidez: aunque la casa no limita la cantidad a apostar, depende de los usuarios elevar la liquidez de las mismas.

- Tipos de apuesta: la variedad de apuestas es menor en contraste con una casa de apuestas tradicional.

2.2.2 Apuestas en el tenis

Como se expuso en apartados anteriores este proyecto se centró en el deporte del tenis. Una de los sistemas de apuestas más conocidos y aplicados, en la actualidad, en el tenis es el *trading*.

Debido a la similitud de las casas de intercambio de apuestas con el mercado de valores, algunas de las técnicas utilizadas en el mundo bursátil se pueden utilizar en el mundo de las apuestas deportivas, como es el caso del *trading*. El *trading* consiste básicamente en aprovecharse de las tendencias del mercado para especular con las cuotas. *Trade* significa intercambio. En el ámbito de las apuestas significa comprar en

un partido o evento deportivo una cuota (apostar a favor) y venderla cuando la cuota haya bajado (apostar en contra o apostar por el otro participante) y obtener un beneficio sin esperar al final del partido independientemente de quién sea el ganador.

El tenis es uno de los deportes más apropiados para el *trading* porque las cuotas cambian mucho y muy rápidamente. Hoy en día muchos apostantes consiguen grandes beneficios aplicando esta técnica.

Para más información en el siguiente enlace se puede consultar un video tutorial en el que se detalla cómo realizar un *trading* con el sacador un partido de tenis: http://www.misapuestas.es/tutorial-apuestas-trading-videotutorial-trading-al-sacador-en-tenis.

Capítulo 3:
ANÁLISIS DEL PROBLEMA

En este capítulo comienza la descripción el proceso de minería de datos, como ya se ha descrito anteriormente se utilizó para llevarlo a cabo la metodología CRISP-DM. En esta primera fase de CRISP-DM se trató de definir y comprender los objetivos perseguidos con la elaboración de este proyecto para lo cual se definió el problema de minería de datos a tratar y se elaboró un plan preliminar que ayudó a alcanzar los objetivos propuestos.

3.1 Objetivos del negocio

A continuación, se describirán los objetivos perseguidos con la realización de este proyecto. El proyecto surgió de la idea de aplicar técnicas de inteligencia artificial para la predicción de resultados de eventos deportivos y a su vez aplicar esta predicción, para probar la viabilidad de un posible negocio en el mercado de las apuestas deportivas.

Una vez entendida y descrita la idea principal del proyecto se describen cuales son los criterios de éxito del proyecto desde el punto de vista del negocio. En este caso se podría abordar el problema desde dos puntos distintos de vista:

- Uno de los criterios a tener en cuenta a lo largo de la realización del proyecto fue el porcentaje de acierto en la predicción del resultado de un evento, en este caso y como se detallará más adelante el proyecto se centró en partidos de tenis masculino e individual, por tanto cada partido es lo que se consideró como un evento. Hoy en día en el mundo de la apuestas se puede apostar a cualquier cosa y dentro de los partidos de tenis también. Este proyecto únicamente se centró en el ganador de cada partido de tenis, por lo que las predicciones solo podían tener dos posibles resultados, o gana un jugador o gana el otro. Así pues, uno de

los criterios de éxito del proyecto fue aumentar el porcentaje de acierto en la predicción del ganador de un partido. Para mejorar estos resultados se utilizaron a lo largo del proyecto tanto estadísticas de juego de los jugadores implicados en el partido, como datos del partido (tipo de superficie, torneo, ronda, etc) y datos de las cuotas del partido en cuestión.

- En este proyecto se trató de analizar la viabilidad de aplicar técnicas de inteligencia artificial para la predicción de partidos de tenis y a su vez aplicar la predicción al mercado de las apuestas. Por tanto, otro de los criterios a los que hubo que prestar atención fue la ganancia obtenida al realizar las apuestas. Para ello se siguieron distintas estrategias que serán explicadas en detalle en apartados posteriores. Sólo se trabajó sobre los eventos en modo pre-partido, es decir, no se analizaron las apuestas a los eventos una vez hubieran comenzado (en vivo). Para realizar el estudio se utilizaron los datos de las apuestas obtenidos antes del inicio de los partidos y se trabajó con las primeras cuotas a la que se apostó a favor de uno y otro jugador en el evento, las últimas, las máximas, las mínimas y las cuotas medias de cada uno.

3.2 Evaluación de la situación

En este apartado se enunciarán los recursos disponibles para la realización del proyecto ya fueran humanos como de hardware, software u otros. Además también se listarán los riesgos que podían haber retrasado la elaboración del proyecto así como los planes de contingencia correspondientes a cada uno de los riesgos. Por último, se detallará el presupuesto del proyecto.

3.2.1 Recursos disponibles

A continuación, se enumeran y describen los distintos recursos con los que se contó para llevar a cabo la realización del proyecto.

- **Recursos humanos:** en este proyecto el coste de personal fue al que hubo que destinar la mayor parte del presupuesto. En este caso, se entiende por coste de personal el coste derivado de la dedicación de los trabajadores a las distintas actividades del proyecto, estos trabajadores son los recursos humanos disponibles para la elaboración del proyecto. En este proyecto se contó con los siguientes trabajadores:
    - **Jefe de proyecto:** Sus principales funciones fueron llevar a cabo la revisión del proyecto y la consolidación de las conclusiones.

  - **Analista / Experto del dominio:** Debió de conocer a fondo el dominio a tratar, es decir, el mercado de las apuestas deportivas. Llevó a cabo la toma de decisiones acerca de qué experimentos realizar así como la extracción de conclusiones.

  - **Analista / Diseñador:** Fue el encargado del manejo de la herramienta de minería de datos, en este caso Weka (Hall). Por supuesto debía conocer a fondo los diferentes algoritmos de inteligencia artificial incluidos en la herramienta.

  - **Programador:** Fue el encargado de la preparación de los datos y de programar las consultas reclamadas por los analistas para los distintos experimentos.

- **Recursos Hardware:** El tamaño del proyecto no requirió una gran inversión en hardware ya que al no haber muchos trabajadores trabajando en éste no se necesitaron múltiples equipos para el desarrollo del proyecto.

  - **Ordenador personal:** Fue necesario un ordenador personal de un coste medio ya que para la elaboración del proyecto no fue necesaria la compra de un ordenador "último modelo".

  - **Impresora:** A veces era necesaria la impresión en papel de algunos informes, diagramas o esquemas para lo que se necesitó una impresora. No fue necesario un gran gasto en este componente ya que no fueron necesarias impresiones de alta calidad.

- **Recursos Software:** Por recursos software se entiende el pago de las licencias software de todos los productos utilizados para la realización del proyecto.

  - **Windows® 7 Home Premium:** Fue el sistema operativo instalado en el ordenador personal para la realización del proyecto y sobre el que fueron instaladas las distintas herramientas con las que se trabajó a lo largo de la realización del proyecto.

  - **Microsoft Office Professional 2007:** Suite de ofimática que contiene procesador de textos (Word), hojas de cálculo (Excel), programa para el desarrollo de presentaciones (PowerPoint) y un programa sistema de gestión de base de datos relacional (Access). Todos estos elementos fueron necesarios a lo largo de la elaboración del proyecto.

  - **Microsoft Office Project 2007:** Es el software de gestión de proyectos con el que se trabajó sobre todo en la planificación inicial del proyecto.

- **Weka:** Software para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Es un software libre distribuido bajo licencia GNU-GPL (Hall).
- **Eclipse:** Entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (Eclipse, 2001).
- **Antivirus McAfee® Security Centre:** Para garantizar la seguridad del ordenador personal.

- **Datos:** Para la realización del proyecto se utilizaron datos provenientes de dos fuentes distintas.
  - **OnCourt:** es un programa para todos los aficionados al tenis. Contiene información sobre los resultados de más de 420 mil partidos de tenis jugados por los mejores jugadores del mundo desde 1990. El programa incluye tanto tenis masculino como tenis femenino. De este programa se obtuvo información estadística acerca de cualquier jugador de tenis, torneo o partido.
  - **Betfair:** Es la primera empresa de juego dedicada a las apuestas de intercambio entre usuarios. Además de ofrecer el servicio de intercambio de apuestas es posible descargarse de ella los datos de los movimientos de las distintas apuestas intercambiadas.
- **Material fungible:** El material necesario para el desarrollo del proyecto fue el típico material de oficina, es decir, folios, bolígrafos, tóner para la impresora, etc. Dicho material fue necesario para la elaboración de los documentos, para la gestión interna del proyecto y para la entrega final del proyecto.

3.2.2 Riesgos y contingencias

En este apartado se hará una identificación de los riesgos de la forma más exhaustiva posible, para verificar que se tuvieron en cuenta todos los casos posibles de riesgos y se dispuso de un plan de contingencia para cada uno de ellos. De esta manera, se consiguió estar prevenido ante dichas situaciones evitando situaciones peligrosas y minimizando su impacto en la planificación y el coste del proyecto.

A continuación, en la Tabla 2 se muestran los riesgos identificados y los planes de contingencia asociados a los mismos.

Riesgos	Contingencias
Riesgos medioambientales: Dentro de este tipo de riesgos entran todos aquellos producidos por desastres naturales como los huracanes, terremotos, incendios, altas temperaturas, etc.	Hacer copias de seguridad almacenándolas físicamente en lugares distintos al origen. Realizar copias de seguridad utilizando herramientas online.
Insuficiencias tecnológicas: los recursos tecnológicos previstos para la realización del proyecto son insuficientes	Elaborar un plan detallado de las necesidades tecnológicas a lo largo del desarrollo del proyecto.
El equipo utilizado para la realización del proyecto falla en algún momento.	Asegurar la adquisición de un material adecuado y con la suficiente garantía. Realizar copias de seguridad con regularidad y sin que haya más de tres días de trabajo entre copia y copia.
Fallo del software utilizado.	Adquirir software de calidad y suficientemente probado.
Puede producirse una mala estimación del tiempo que se dedicará a cada una de las tareas/actividades. El problema surgiría cuando se produjera algún retraso en alguna de ellas.	Afinar en la estimación del esfuerzo requerido para cada una de las tareas/actividades y prevenir posibles problemas y retrasos dejando un margen de tiempo razonable para solucionarlos.
Identificar todas las tareas/actividades necesarias para llevar a cabo el proyecto.	Analizar detenidamente los requisitos del proyecto y estudiar la metodología que se va a seguir para la elaboración del proyecto y de esta forma planificar todas las tareas/actividades necesarias en el proyecto.
La naturaleza de los datos no es adecuada para utilizar técnicas de minería de datos.	Analizar la viabilidad de la solución en una instancia inicial del proyecto.

Tabla 2: Riesgos y contingencias del proyecto.

3.2.3 Presupuesto

En este apartado se van a detallar y explicar todos los costes que tuvo asociados el proyecto. Los costes estarán agrupados por categorías para lograr una mayor claridad y comprensión de los mismos.

Costes de personal

Estos costes se refieren a los salarios de todos los integrantes del grupo de desarrollo, durante el periodo de desarrollo del proyecto. El salario de cada miembro

está definido por el rol que desempeñó en el proyecto y por las horas de trabajo. Dependiendo del rol, la hora de trabajo costó más o menos que la de otro miembro del grupo con un rol diferente. El coste de personal constituyó la mayor parte del coste total del proyecto. El primer paso fue determinar el coste por hora de cada uno de los cuatro profesionales que participaron en el desarrollo del proyecto. A continuación, en la Tabla 3 se detallan los sueldos por hora de cada uno de ellos.

Rol	Sueldo por hora (€/h)
Jefe de proyecto	75
Analista / Experto en el dominio	66
Analista / Diseñador	60
Programador	40

Tabla 3: Sueldo por hora de los diferentes profesionales implicados en el proyecto.

El siguiente paso fue determinar las distintas actividades a realizar en el proyecto junto con su duración correspondiente. Esta información se detalla en la Tabla 4.

Nº	Actividad	Duración (horas)
1	Comprensión del dominio	10
2	Identificación de los objetivos	15
3	Planificación	5
4	Comprensión de los datos	30
5	Comprensión de las herramientas	15
6	Preparación de los datos	120
7	Diseño del plan de experimentos	30
8	Experimentación	300
9	Recopilación de resultados	50
10	Análisis de los resultados	80
11	Extracción de conclusiones	40
12	Documentación del proyecto	150
13	Preparación de la presentación del proyecto	20
		865

Tabla 4: Relación de actividades del proyecto y duración de las mismas.

Cada una de las distintas actividades no tuvo por qué ser realizada completamente por un único profesional de los implicados en el proyecto por lo que a continuación se detalla el nivel de implicación que tuvo cada profesional en cada una de las actividades.

Actividad	Profesional	% de implicación	Horas dedicadas
1	Jefe de proyecto	30	3
	Analista / Experto en el dominio	10	3
	Analista / Diseñador	30	1
	Programador	30	3
2	Jefe de proyecto	50	7,5
	Analista / Experto en el dominio	30	4,5
	Analista / Diseñador	20	3
	Programador	0	0
3	Jefe de proyecto	60	3
	Analista / Experto en el dominio	20	1
	Analista / Diseñador	20	1
	Programador	0	0
4	Jefe de proyecto	25	7,5
	Analista / Experto en el dominio	25	7,5
	Analista / Diseñador	25	7,5
	Programador	25	7,5
5	Jefe de proyecto	30	4,5
	Analista / Experto en el dominio	30	4,5
	Analista / Diseñador	10	1,5
	Programador	30	4,5
6	Jefe de proyecto	10	12
	Analista / Experto en el dominio	10	12
	Analista / Diseñador	10	12
	Programador	70	84

Actividad	Profesional	% de implicación	Horas dedicadas
7	Jefe de proyecto	10	3
	Analista / Experto en el dominio	80	24
	Analista / Diseñador	10	3
	Programador	0	0
8	Jefe de proyecto	5	15
	Analista / Experto en el dominio	40	120
	Analista / Diseñador	40	120
	Programador	15	45
9	Jefe de proyecto	10	5
	Analista / Experto en el dominio	45	22,5
	Analista / Diseñador	45	22,5
	Programador	0	0
10	Jefe de proyecto	20	16
	Analista / Experto en el dominio	40	32
	Analista / Diseñador	40	32
	Programador	0	0
11	Jefe de proyecto	20	8
	Analista / Experto en el dominio	70	28
	Analista / Diseñador	10	4
	Programador	0	0
12	Jefe de proyecto	10	15
	Analista / Experto en el dominio	60	90
	Analista / Diseñador	20	30
	Programador	10	15
13	Jefe de proyecto	70	14
	Analista / Experto en el dominio	10	2
	Analista / Diseñador	10	2

Actividad	Profesional	% de implicación	Horas dedicadas
	Programador	10	2

Tabla 5: Asignación de actividades por roles y cálculo de horas dedicadas.

A continuación, se han de sumar las horas de cada uno de los roles que intervinieron en el desarrollo del proyecto para de esta forma poder calcular el coste del personal multiplicando las horas invertidas por cada rol por el salario por hora de los mismos.

Rol	Sueldo por hora (€/h)	Horas invertidas	Coste Total (€)
Jefe de proyecto	75	113,5	8.512,5
Analista / Experto en el dominio	66	351	23.166
Analista / Diseñador	60	239,5	14.370
Programador	40	161	6.440
			52.488,5

Tabla 6: Recopilación de horas y costes por rol del personal.

Costes de hardware

A continuación, en la Tabla 7 se detallan los costes relacionados con los componentes hardware necesarios para el desarrollo del proyecto. También se incluyen en ella los precios totales de cada equipo. La vida útil de los equipos informáticos es de tres años, por lo que anualmente se amortiza un tercio de su coste total.

La parte de amortización imputada al proyecto es la proporcional a la duración del mismo, que según los cálculos realizados en el apartado anterior puede estimarse en algo más de cinco meses de dedicación al proyecto, para estos cálculos se consideró un mes como veinte días trabajados a ocho horas de trabajo diario[1]. Por tanto, la parte del coste que se asumió en este proyecto fue la amortizada durante la realización de este proyecto, considerando como se ha dicho antes que la amortización total de los productos se lleva a cabo en treinta y seis meses.

[1] Realmente el trabajo se ha llevado a cabo a lo largo de ocho meses naturales

Concepto	Cantidad	Precio unitario (€)	Coste Total (€)
Ordenador personal	1	776	776
Impresora	1	57,12	57,12
			833,12

Tabla 7: Costes del hardware.

Como se ha comentado antes la duración estimada del desarrollo del proyecto fue de cinco meses y el tiempo de amortización del hardware es de 36 meses, por tanto el coste hardware asociado al proyecto fue de **115,71 €.**

Las características de los componentes hardware son las listadas a continuación:

- **Ordenador personal:** Dell Inspiron 560s, Procesador Intel® Core™ 2 Q8200s de núcleo cuádruple (2,33 GHz, FSB a 1.333 MHz, caché de 4 MB), Windows® 7 Home Premium original 64bit – España, Tarjeta gráfica Nvidia ® GeForce G310 de 512 MB, Memoria DDR3 de doble canal a 1.066 MHz de 6.144 MB [2x2048 + 2x1024], Disco duro SATA de 1 TB (7.200 rpm), Unidad óptica de DVD+/-RW a 16X (lectura y escritura de DVD/CD) (sólo Windows 7), 1 año de cobertura incluido con su ordenador, McAfee® Security Centre -15 meses de protección– Español.
- **Impresora:** Impresora multifunción de inyección de tinta en color Dell V105. Multifunción impresora/copiadora/escáner, conectividad USB 2.0 de alta velocidad (conector tipo B), hasta 4.800 x 1.200 ppp para una impresión y color precisos.

Costes de software

A continuación, se detallan los costes asociados al software necesario para desarrollar el proyecto. A diferencia del hardware, el software se compra mediante licencias, es decir, permisos para utilizar un cierto software en un número determinado de ordenadores durante un periodo de tiempo. En algunos casos, dicho periodo de tiempo es ilimitado.

Tanto el antivirus como el sistema operativo para el ordenador personal necesitan que se adquieran sus correspondientes licencias para poder hacer uso de ellos, en este caso las licencias de ambos están incluidas en el coste del ordenador personal, por lo que su coste no viene reflejado en este apartado.

Como la licencia del antivirus es por 15 meses fue suficiente para la realización del proyecto. La licencia del Sistema Operativo no tiene duración limitada y al estar su coste incluido en el del ordenador personal se consideró tres años el tiempo de amortización de dicha licencia.

Las dos licencias de Microsoft Office también son ilimitadas pero se consideró su periodo de amortización de tres años.

Concepto	Cantidad	Precio unitario (€)	Coste Total (€)
Windows® 7 Home Premium	1	0 (incluido en PC)	0
Microsoft Office Professional 2007	1	595,56	595,56
Microsoft Office Project 2007	1	654,36	654,36
McAfee® Security Centre	1	0 (incluido en PC)	0
Weka	1	0 (software libre)	0
Eclipse	1	0 (software libre)	0
			1.249,92

Tabla 8: Costes del software.

Tanto Weka como Eclipse son software libre por lo que no representaron ningún coste en el proyecto. Como ocurría con los costes hardware, para calcular el coste del software del proyecto hubo que tener en cuenta que la duración del desarrollo del proyecto, que fue de cinco meses siendo el tiempo de amortización del software de 36 meses. Por tanto el coste software asociado al proyecto fue de **173,6 €.**

Costes del material fungible

En este apartado se detallan los costes relacionados con el material fungible. Al igual que en el apartado anterior (Costes de software) se incluyen los precios totales de cada material.

El material fungible necesario para el desarrollo del proyecto fue el típico material de oficina, es decir, folios, bolígrafos, tinta para la impresora, etc. Dicho material fue necesario para la elaboración de los documentos, para la gestión interna del proyecto y para la entrega final del proyecto.

A continuación, en la Tabla 9 se muestran los costes del material fungible asociados al proyecto:

Concepto	Cantidad	Precio unitario (€)	Coste Total (€)
Folios (Paquete 500)	2	2,52	5,04
Bolígrafo	3	0,2	0,6
CD (tarrina 10)	1	2,09	2,09
DVD (tarrina 10)	1	2,93	2,93
Tinta impresora	2	20,37	40,74
			51,4

Tabla 9: Costes material fungible.

Costes de los gastos indirectos

Los gastos indirectos del proyecto se calculan como el 10% sobre el total del resto de los costes. En este caso incluyeron principalmente los siguientes gastos:

- Suministro eléctrico con la compañía Iberdrola durante los cinco meses que duró el desarrollo del proyecto. Con esto lo que se persiguió es poder hacer uso de las herramientas necesarias para llevar a cabo dicho proyecto.
- Contratación de una línea telefónica para poder contratar a su vez una conexión a internet del tipo ADSL que garantizase una conexión estable a internet y así poder disfrutar de todas sus ventajas.
- Mantenimiento de las infraestructuras utilizadas para la realización del proyecto.

Sumando los costes anteriores se obtuvo un resultado de 52.829,21 por tanto, el coste total de los gastos indirectos fue de **5.282,92 €.**

Coste total

Dada la naturaleza del proyecto no se consideró necesario ni oportuno incluir ningún porcentaje de beneficio sobre el coste total del mismo, si sería necesario incluirlo en un supuesto presupuesto a presentar a un posible cliente. Tampoco se tuvo en cuenta el IVA en los costes presentados.

A continuación, en la Tabla 10 se resumen los gastos del proyecto desglosados en los conceptos anteriormente descritos siendo el coste total del proyecto de **58.112,13 €.**

Concepto	Coste Total (€)
Costes de personal	52.488,5
Costes de hardware (sólo lo amortizado a lo largo del proyecto)	115,71
Costes de software (sólo lo amortizado a lo largo del proyecto)	173,6
Costes del material fungible	51,4
Costes de gastos indirectos (10% de la suma de lo anterior)	5.282,92
	58.112,13

Tabla 10: Resumen de costes del presupuesto.

3.2.4 Cronograma del proyecto

A continuación, se muestra el cronograma del proyecto, para ello se elaboró un diagrama de Gantt en el que se muestran las distintas actividades de las que se compuso el proyecto y los recursos empleados para la realización de cada una de ellas.

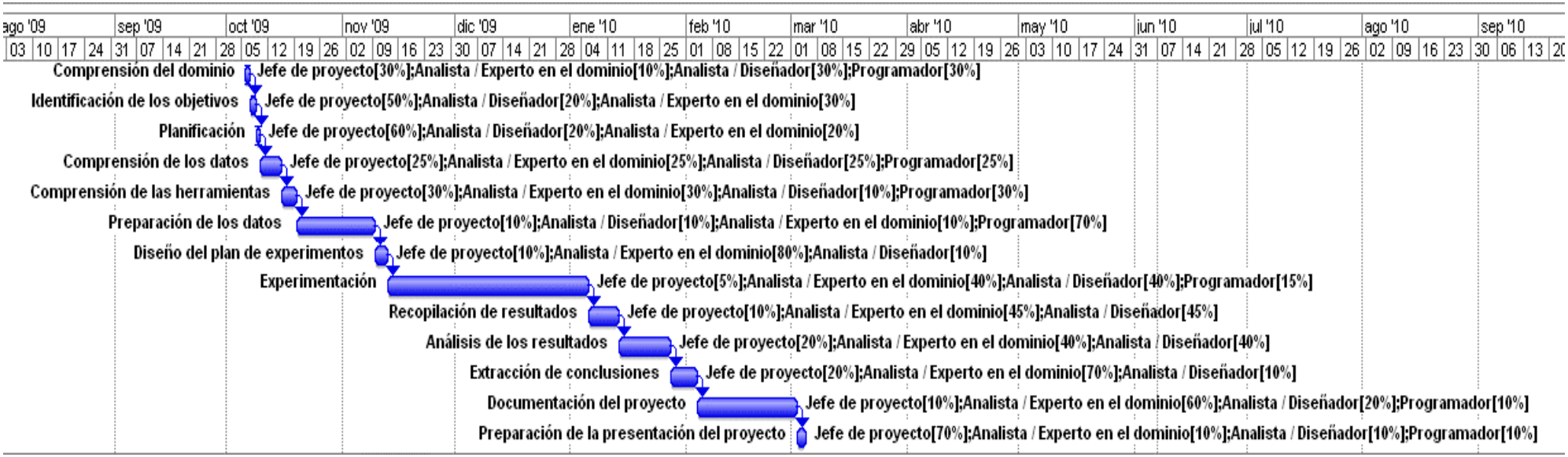


Figura 3: Cronograma del proyecto en formato diagrama de Gantt.

3.3 Objetivos de la minería de datos

La minería de datos puede definirse como la extracción no trivial de información implícita, previamente desconocida y potencialmente útil, a partir de los datos (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). Para conseguirlo hace uso de diferentes tecnologías que resuelven problemas típicos de agrupamiento automático, clasificación, asociación de atributos y detección de patrones secuenciales. En este apartado se tratan de explicar los objetivos que fueron perseguidos en este proyecto con la aplicación de la minería de datos.

El objetivo final de cualquier proyecto de minería de datos puede resumirse en uno de estos dos:

- Ahorrar dinero mejorando la eficacia de sus actividades.
- Ganar dinero descubriendo nuevas fuentes de beneficios.

El objetivo de este proyecto de no ser meramente académico estaría claramente encuadrado en la segunda de las opciones. El objetivo de la minería de datos para este proyecto fue a partir de un conjunto de datos, en este caso estadísticas de partidos de tenis y los movimientos de las apuestas en estos partidos, y un conjunto de técnicas, las ofrecidas por la herramienta Weka, tratar de llegar a unas conclusiones que permitiesen diseñar un método ganador a largo plazo en el mundo de las apuestas deportivas.

En este caso los objetivos que se persiguieron para encontrar un método ganador fueron principalmente aumentar el porcentaje de acierto del jugador ganador de un partido de tenis, predecir el momento óptimo para realizar una apuesta y analizar la posibilidad de aplicar estrategias de *trading*. Por tanto, aplicando la minería de datos se trató de conseguir árboles de decisión o conjuntos de reglas que llevasen a predecir el ganador de un partido de tenis con un gran porcentaje de éxito. Una vez hecha esta predicción se analizaron los datos para averiguar cuál era el momento óptimo, entendiendo por óptimo con la cuota más alta, para realizar una apuesta a favor de la predicción dada.

Debido a la gran cantidad de datos disponibles, fue necesario aplicar técnicas de selección de atributos que ayudasen a mejorar los resultados eliminando atributos innecesarios o que aportasen muy poca información para la predicción de los resultados de los partidos. También se aplicaron técnicas de asociación para intentar identificar las interrelaciones más importantes entre los distintos atributos.

Por tanto, y a modo de resumen, a lo largo del proyecto se usaron técnicas de asociación, clasificación y selección de atributos para tratar de aumentar la ganancia de una posible inversión en el mundo de las apuestas.

3.4 Plan del proyecto

En este apartado se describe el plan para alcanzar los objetivos de la minería de datos y con ellos los del negocio. Para que su visualización resulte más cómoda se muestra la planificación en la Tabla 12 incluyendo el nombre de las etapas ejecutadas, juntos con su duración, recursos requeridos, herramientas, técnicas, entradas, salidas, y dependencias.

La Tabla 11 describe el identificador asignado a cada recurso:

Rol	Identificador
Jefe de proyecto	1
Analista / Experto en el dominio	2
Analista / Diseñador	3
Programador	4

Tabla 11: Correspondencia rol-identificador

Los recursos requeridos van en porcentaje teniendo que sumar entre todos el 100%. A continuación, se detalla en la Tabla 12 el plan de proyecto.

Tarea	Duración (horas)	Dependencias	Recursos				Herramientas	Técnicas	Entradas	Salidas
			1	2	3	4				
Comprensión del dominio	10		30	10	30	30				
Identificación de los objetivos	15	1	50	30	20	0	Word			Objetivos del proyecto
Planificación	5	2	60	20	20	0	Project	CRISP-DM	Objetivos del proyecto	Planificación del proyecto
Comprensión de los datos	30	3	25	25	25	25	Access		Datos estadísticos y de apuestas	Informe con el significado de cada tabla/atributo/campo
Comprensión de las herramientas	15	4	30	30	10	30	Weka, Access, Word, Excel, Project, Eclipse, PowerPoint		Herramientas	
Preparación de los datos	120	5	10	10	10	70	Eclipse, Access, Word		Datos estadísticos y de apuestas	Datos modificados
Diseño del plan de experimentos	30	6	10	80	10	0	Word, Weka, Access		Datos, herramientas, objetivos, planificación	Planificación de los experimentos del proyecto
Experimentación	300	7	5	40	40	15	Weka	Asociación, Clasificación, Selección de atributos	Datos limpios, planificación de los experimentos	Resultados experimentos
Recopilación de resultados	50	8	10	45	45	0	Weka, Word, Excel		Resultados	Hoja Excel con el resumen de los resultados
Análisis de los resultados	80	9	20	40	40	0	Excel		Resumen de los resultados	Informe con el conocimiento adquirido
Extracción de conclusiones	40	10	20	70	10	0	Word, Excel		Informe con el conocimiento adquirido de los resultados	Informe con las conclusiones del proyecto
Documentación del proyecto	150	11	10	60	20	10	Word		Todo lo anterior	Memoria del proyecto
Preparación de la presentación	20	12	70	10	10	10	PowerPoint		Memoria del proyecto	Presentación PowerPoint

Tabla 12: Plan del proyecto.

Capítulo 4:
COMPRENSIÓN DE LOS DATOS

En este capítulo se presentan las distintas tareas realizadas con el objetivo de familiarizarse con los datos para, de esta forma, comprender el problema y tener los conocimientos suficientes para afrontar con garantías la siguiente fase, que no es otra que la preparación de los datos.

4.1 Recolección inicial de datos

El primer paso, una vez que se decidió el deporte en el que se centró el proyecto, fue buscar los datos para la elaboración del mismo. Hoy en día y gracias a internet se dispone de numerosos medios que ofrecen estadísticas de jugadores y partidos de tenis. Después de realizar una búsqueda exhaustiva en la red se obtuvo un listado de numerosas páginas webs y programas que ofrecían los datos requeridos. En este caso, se decidió que se debía escoger una web o programa y centrarse únicamente en ella, ya que, si hubiera que trabajar con varias fuentes se complicaría demasiado la tarea de formar la base de datos, pues no es lo mismo trabajar con una única fuente de datos que formar la base de datos juntando datos de distintas fuentes, ya que, esta segunda opción aumentaría de forma considerable la carga de trabajo al tener que unificar todo para formar una base de datos única.

Por tanto, una vez analizados los pros y los contras de cada una de las distintas alternativas, se decidió trabajar con la base de datos ofrecida por el programa *OnCourt* (OnCourt). *OnCourt* es un programa para los aficionados al tenis que contiene información sobre los resultados de más de 420.000 partidos de tenis, jugados por los mejores jugadores del mundo desde 1990. El programa incluye tanto datos del tenis masculino como del femenino. Con este programa se puede obtener una gran cantidad de información estadística acerca de cualquier jugador de tenis, de cualquier torneo de tenis o el historial de los enfrentamientos cara a cara entre cualquier par de jugadores.

Para obtener los datos se procedió a la descarga del software en el ordenador personal y a su posterior instalación. Una vez instalado el software, en la carpeta en la que se realizó dicha instalación se almacena la base de datos inicial que sirvió como uno de los dos puntos de partida del proyecto, en cuanto a datos se refiere. El sistema gestor de la base de datos es Microsoft Access que fue el gestor utilizado a los largo del proyecto.

El otro punto de partida fueron los datos ofrecidos por *Betfair* (Betfair). *Betfair* fue la primera empresa de juego dedicada a las apuestas de intercambio entre usuarios. En este momento es la compañía más grande en el ámbito del intercambio de apuestas en Internet. La fórmula del intercambio de apuestas permite a los usuarios apostar a cuotas marcadas por otros usuarios, en lugar de depender de un corredor de apuestas. En *Betfair*, los usuarios pueden apostar "A favor" (apuestas normales a una selección ganadora) o "En contra" (apuestas en contra de un resultado), eliminando así la necesidad de un corredor de apuestas como intermediario. *Betfair* cobra una comisión por apuesta ganada. La comisión estándar es del 5%, pero puede reducirse hasta un 2% en función a las cantidades que apueste el usuario en el sitio Web. De esta forma, *Betfair* no penaliza las apuestas falladas y promueve las apuestas ganadoras. Además, se proporciona al usuario toda la información disponible para analizar mejor los posibles resultados con estadísticas, foros y un blog de noticias. Parte de esta información fue utilizada para la elaboración de este proyecto. Así pues, accediendo a la página web http://data.betfair.com/ se obtuvieron los datos de todas las apuestas realizadas en *Betfair* desde junio de 2004.

4.2 Descripción de los datos iniciales

En este apartado se procede a describir los datos adquiridos en su formato original. Como es normal estos datos tuvieron que ser tratados para poder formar con ellos una base de datos coherente y consistente con la que se pudiera trabajar a lo largo del proyecto. Como se ha descrito anteriormente los datos fueron obtenidos de dos fuentes distintas, *OnCourt* y *Betfair*. A continuación, se muestran los datos de cada una de ellas.

4.2.1 Datos de OnCourt

En la base de datos que se obtuvo al instalar el software *OnCourt* se pudo acceder a las tablas que se muestran en la Figura 4:

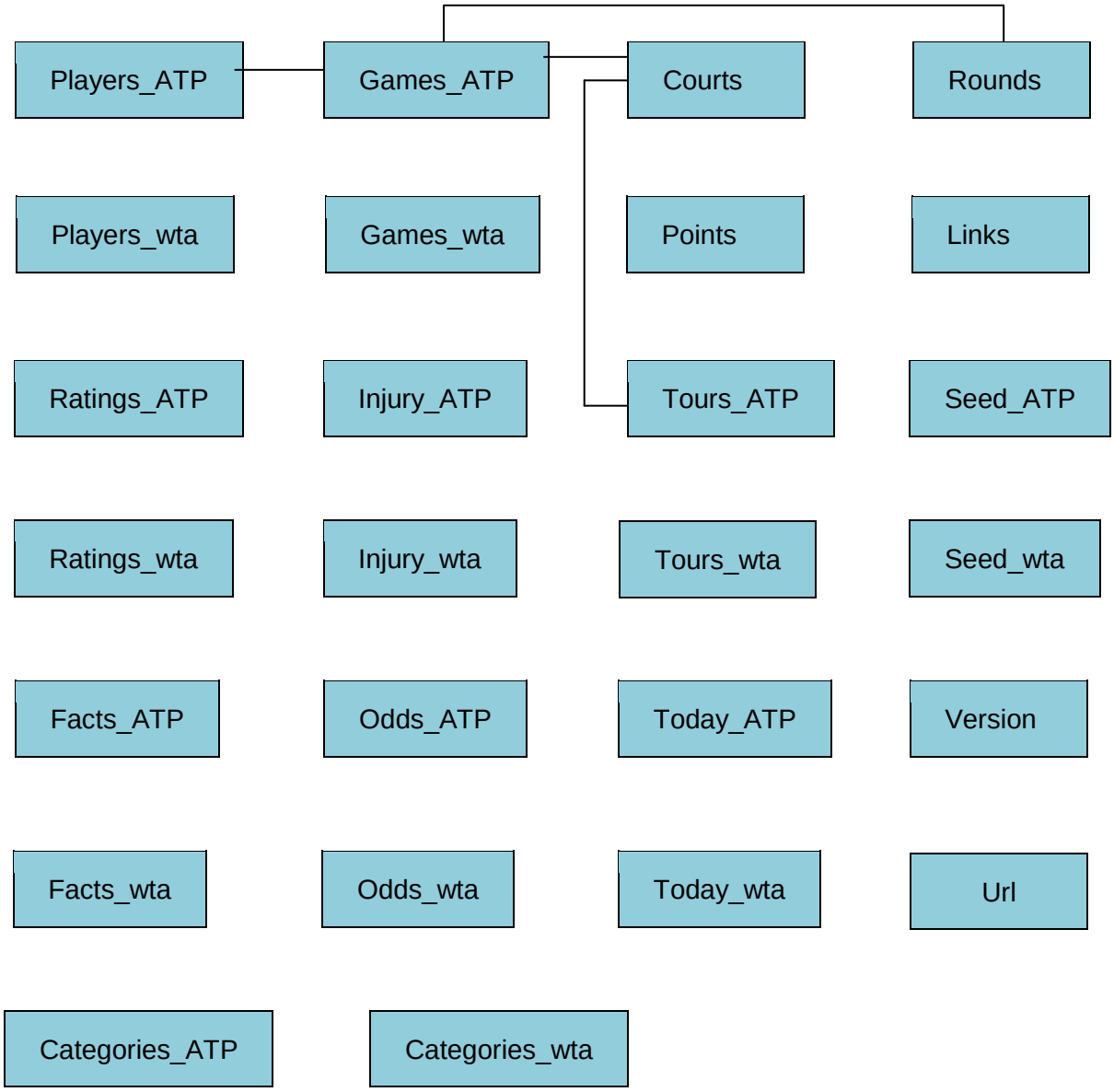


Figura 4: Tablas de la base de datos original de *OnCourt*.

Como se pueden observar las tablas de la base de datos original no estaban en su mayoría relacionadas entre sí, esto es algo que hubo que corregir puesto que a simple vista se observa que la mayoría de tablas deberían de están relacionadas a través de relaciones como las siguientes:

- Un partido lo juegan dos jugadores.
- Un partido pertenece a un torneo.
- Un partido se juega en una ronda.
- Un jugador es o no es cabeza de serie en un torneo.

- Un jugador tiene unas estadísticas totales.
- Un jugador genera unas estadísticas en un partido.

Por tanto, la ausencia de relaciones en la base de datos original es algo que debió corregirse.

A poco que se sea conocer del mundo del tenis y se conozcan las asociaciones de tenistas profesionales, tanto masculina (ATP) como femenina (WTA), se puede detectar a simple vista una clara distinción entre algunas tablas que sólo almacenan datos de tenis femenino y otras que sólo los almacenan del masculino, esto será de gran ayuda puesto que el proyecto elaborado sólo analizó datos del tenis masculino como ya se avanzó anteriormente. De esta forma, las tablas en cuyo nombre se pueda encontrar la palabra “wta” almacenaban sólo datos del tenis femenino mientras que, las tablas que contengan la palabra “atp” sólo almacenaban datos del tenis masculino.

Los atributos de las tablas son de tipo texto, numérico o fecha/hora por lo que no hubo que trabajar con tipos complejos de datos. La base de datos inicial actualizada a día nueve de marzo de dos mil nueve ocupaba un total de 52.1 MB.

Se puede observar a simple vista que las necesidades del proyecto estaban cubiertas, ya que, se contaba con datos de los jugadores, partidos, torneos, rondas, puntuaciones, históricos de puntuaciones, estadísticas e incluso datos de apuestas de otras casas distintas a *Betfair*.

4.2.2 Datos de Betfair

Los datos para la realización del proyecto fueron obtenidos a finales de marzo del año dos mil nueve. Para obtener los datos de http://data.betfair.com/ hay que cumplir dos requisitos, el primero es estar registrado en http://www.betfair.com/ y el segundo es tener una cantidad mínima de 100 puntos *Betfair*. Estos puntos se consiguen haciendo uso de los servicios ofrecidos por *Betfair* (intercambio de apuestas, póquer, etc). *Betfair* actualmente actualiza los datos cada semana.

Los datos ofrecidos por *Betfair* son divididos en dos conjuntos, datos de carreras de caballos y datos de otros eventos (en su mayoría deportes). En este caso sólo se obtuvieron los datos del segundo conjunto pues es en el que se encuentran los datos pertenecientes al tenis. El tamaño total de los datos del conjunto seleccionado fue de 13.5 GB. Es evidente que el tamaño total de los datos pertenecientes al tenis masculino era infinitamente inferior y será detallado en apartados posteriores. *Betfair* ofrece los datos en archivos de valores separados por comas (csv) a excepción de los datos del año 2004 (primer año en que fueron publicados) que se presentan en un fichero de texto separando los datos por el carácter tabulador.

Estos datos se obtienen a través de varios ficheros por cada uno de los años. En cada fichero la primera línea está formada por el título de las columnas y las siguientes líneas son los datos.

A continuación, se detalla la descripción de los datos almacenados en cada una de las columnas.

- `EVENT_ID`: Identificador del evento al que pertenece la apuesta. *Betfair* identifica unívocamente los eventos que ofrece para apostar, como evento en este caso se entiende cada partido de tenis.

- `FULL_DESCRIPTION`: Descripción del evento. Este campo es irregular y varía su formato a lo largo de los años y eventos. Normalmente se encuentra en él información del torneo, la ronda del torneo y los jugadores que juegan el partido al que pertenece la apuesta de la siguiente forma: `Torneo/Ronda/Jugador1 v Jugador2`.

- `SCHEDULED_OFF`: Fecha y hora prevista para el inicio del evento. En todos los torneos de tenis se marca un horario para los partidos pero este horario no siempre se cumple debido a las inclemencias climatológicas, a una duración mayor a la prevista de partidos previos o a otras causas.

- `EVENT`: Indica una pequeña descripción de a qué se está apostando en el evento. El valor del campo varía a lo largo de los años pero por lo general en el campo se indica si la apuesta es a ganador del partido, ganador del set, ganador con hándicap, etc.

- `ACTUAL_OFF`: Fecha y hora real en la que el evento se inició.

- `SELECTION`: Dentro del evento y de la apuesta la selección a favor de la cual se hace la apuesta, es decir, si la apuesta es a ganador de un partido, la selección es el jugador de los que está jugando el partido a favor del cual se apostó.

- `SETTLED_DATE`: Fecha y hora en la que se cerró el evento, es decir, fecha y hora en la que se acaba el partido en cuestión.

- `ODDS`: Cuota a la que se apostó por la selección en el evento.

- `LATEST_TAKEN`: Fecha y hora en la que se realizó la última apuesta a favor de la selección para el evento anterior a la cuota anterior.

- `FIRST_TAKEN`: Fecha y hora en la que se realizó la primera apuesta a favor de la selección y a la cuota anterior.

- `IN_PLAY`: Este campo sólo puede tener tres valores distintos que identifican el tipo de apuesta:

    - `IP` (*In-Play*): Señala que la apuesta fue realizada en vivo, es decir, con el partido ya comenzado.

  - `PE` (*Pre-Event*): Señala que la apuesta fue realizada antes del comienzo del evento/partido siendo posible apostar en vivo en el evento.
  - `NI` (*Event did not go in-play*): En el evento no era posible apostar en vivo por lo que la apuesta fue anterior al comienzo del mismo.

- `NUMBER_BETS`: Número de apuestas que se han realizado en el evento a favor de la selección y a la cuota anterior.
- `VOLUME_MATCHED`: Suma total del volumen cruzado en el evento a favor de la selección y a la cuota anterior.
- `SPORTS_ID`: Identificador del deporte al que pertenece el evento. La correspondencia del valor con cada uno de los deportes puede consultarse en http://data.betfair.com/sportids.htm.
- `SELECTION_ID`: Identificador de la apuesta. Cada evento con su tipo de apuesta y su selección tiene un identificador.
- `WIN_FLAG`: Identificador de apuesta ganada o perdida con dos posibles valores:
  - `1`: Si la selección resultó ganadora total o parcialmente.
  - `0`: En otro caso.

Una vez obtenidos los datos se analizaron para saber si son válidos para el proyecto. Al recorrer los datos se pudo observar que por cada evento hay numerosas líneas con datos de apuestas a ese evento, distintas cuotas, fechas de apuestas, etc. Por tanto, en principio parecía razonable pensar que los datos fuesen útiles para el proyecto pues se podría obtener de cada partido su cuota máxima, mínima y media, la última cuota a la que se apostó antes del inicio del evento o la primera cuota a la que se apostó. Una vez obtenidos y calculados estos datos fueron útiles para realizar predicciones y analizar los índices de una posible ganancia.

4.3 Verificación de la calidad de los datos

En este apartado se examina la calidad de los datos tratando de analizar si están completos, son correctos o tienen errores y en caso de tenerlos cómo de frecuentes son.

Como punto de partida se analizó la base de datos ofrecida por *OnCourt*. *OnCourt* es un software consolidado y el cual concede una versión de prueba de quince días para posteriormente poder obtener la licencia de la versión comercial. Simplemente el hecho de que sea un software de pago ofrece unas garantías en

cuanto a que la base de datos sea fiable y no contenga errores. En cualquier caso, al analizar la base de datos se comprobó que así era, al recorrer todas las tablas de la base de datos y hacer un análisis[1] todas las filas estaban completas, los datos eran coherentes y no existían errores. Para llevar a cabo estas comprobaciones se utilizó la interfaz de usuario del propio software, la base de datos que se utilizó en el proyecto y los datos ofrecidos por la ATP en su página web http://www.ATPworldtour.com. Con las tres fuentes de datos se comprobó que los datos ofrecidos por la interfaz del software eran los almacenados en la base de datos y que estos a su vez eran correctos ya que coincidían con los datos de la ATP, que es el organismo oficial que rige los torneos de tenis masculino.

Posteriormente, se analizaron los datos obtenidos de Betfair. Betfair advierte que no ofrece ningún tipo de garantía de exactitud en los datos. Esto era una advertencia de lo que iba a suceder en el análisis de la calidad. donde se comprobó que existían fallos en los datos. Algunos de los fallos encontrados fueron:

- Error en el nombre de jugadores del tipo: `R. Nadal -> R. Nadl`. De estos errores se pudo deducir que los datos habían sido insertados manualmente por lo que al insertarlos en la base de datos del proyecto habrá que controlar estos errores.
- Error en la descripción del evento: Este tipo de error se producía al insertar la descripción de otro evento que además solía ser del mismo día. No se encontraron muchos pero sí eran errores llamativos.
- Error en el identificador del deporte: Fueron escasos los errores encontrados de este tipo. Simplemente al aplicar el filtro para seleccionar las líneas con datos de tenis, algunas de estas líneas (muy pocas) eran de otros deportes. Esto fue fácil de verificar pues en la descripción aparecían torneos de fútbol, baloncesto, equipos u otros datos.
- Errores en las fechas: Estos errores se corrigieron fácilmente pues se detectaron comprobando los campos con fechas. Son del tipo:
    - Fecha de la primera apuesta era posterior a la de la última, cosa que es imposible en la realidad.
    - Fechas de apuestas posteriores a la fecha de cierre del evento, algo que también es imposible porque tampoco nadie apuesta después de que un evento haya acabado.

[1] El análisis realizado no fue completo pues la cantidad de filas era muy grande.

  - Fechas de apuestas en vivo anteriores al inicio del evento. También es imposible puesto que si el evento no se ha iniciado es imposible que la apuesta haya sido en vivo.

  - Fecha de cierre del evento anterior a la de inicio. Obviamente, es algo imposible, algo que no ha empezado no puede haber terminado.

- Error en las selecciones: Es decir, en un partido seleccionan como ganador a un jugador que no lo jugó.

Todos estos errores fueron encontrados analizando los datos, por lo que, al generar la base de datos que se utilizó a lo largo del proyecto hubo que tenerlos en cuenta para que no derivasen en futuros problemas.

Aunque parezca que había numerosos errores en estos datos, en realidad, esto no fue así, ya que, había más de cuatro millones de líneas/filas sólo de tenis masculino individual y realmente no se encontraron entre ellas más de diez mil líneas erróneas. Además, los errores solían ir en grupos, pues las líneas pertenecientes al mismo evento normalmente estaban todas seguidas y si una de ellas, por ejemplo, contenía un error en la descripción, el resto también lo contenía.

Capítulo 5:
PREPARACIÓN DE LOS DATOS

En este capítulo se describirán las actividades en las que se formó el conjunto final de datos, es decir, los datos que se utilizaron en el proceso de minería de datos. Hasta llegar a la base de datos final las tareas descritas en este capítulo tuvieron que ser realizadas en varias ocasiones.

5.1 Selección de los datos

En el Capítulo 4 se presentaron los conjuntos de datos iniciales a partir de los cuales se formó la base de datos final. A continuación, se detallan los datos que fueron seleccionados y los que fueron excluidos. Como se describió, en apartados anteriores, el dominio del problema estaba centrado en el tenis masculino e individual, por tanto, el primer filtro que se debió aplicar fue el que eliminara los datos que no pertenecían a dicho dominio.

Partiendo de la base de datos de *OnCourt* el filtro más fácilmente aplicable fue el eliminar todas las tablas que contuvieran la palabra `"wta"` pues éstas sólo contenían datos del tenis femenino. Por lo tanto, lo primero fue eliminar las tablas: `Categories_wta`, `Facts_wta`, `Games_wta`, `Injury_wta`, `Odds_wta`, `Players_wta`, `Ratings_wta`, `Seed_wta`, `Stat_wta`, `Today_wta` y `Tours_wta`. También se eliminó la tabla `Version` pues contenía información irrelevante para el desarrollo del proyecto. El resto de tablas contenían información valiosa que fue depurada como se describirá en apartados posteriores. De esta forma, con el filtro aplicado se redujo el número de tablas de la base de datos inicial de *OnCourt*. Las tablas que formaron parte de la base de datos final son las mostradas en la Figura 5. Estas tablas serán detalladas en apartados posteriores.

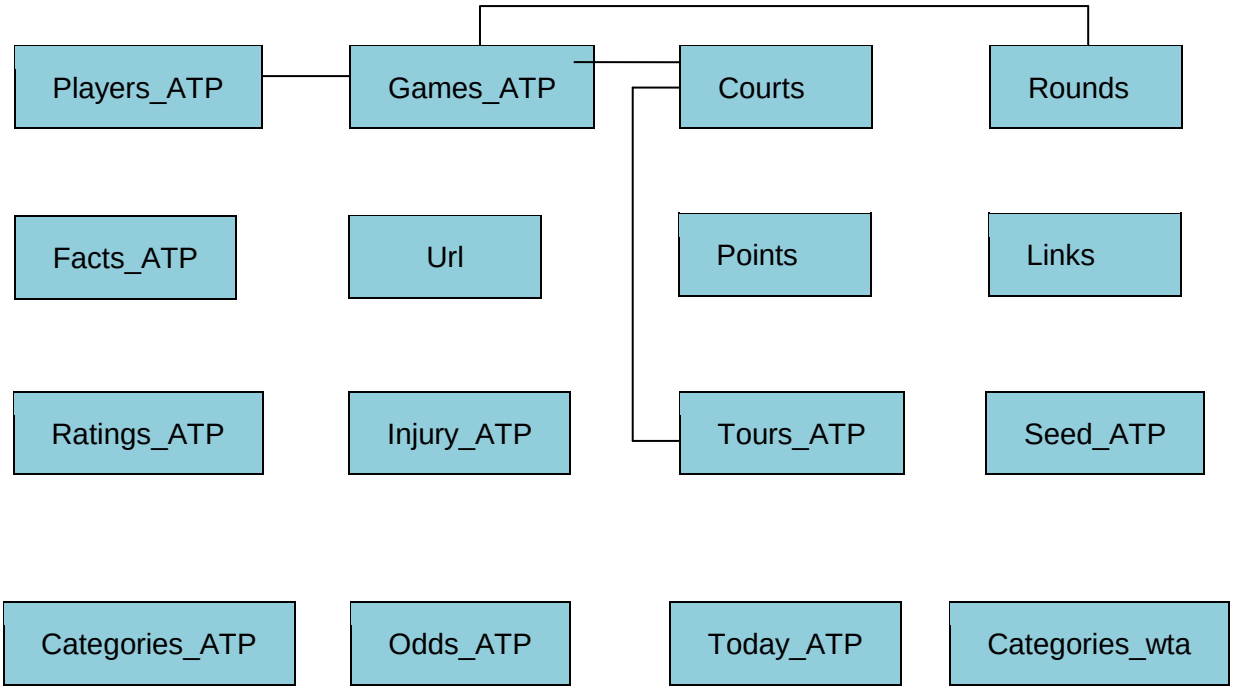


Figura 5: Tablas seleccionadas de la base de datos de OnCourt.

A continuación, se procedió a hacer la selección de los datos de apuestas ofrecidos por *Betfair*. Al empezar a analizar los datos se verificó que los pertenecientes al año 2004 no contaban con la columna de identificador de deporte por lo que complicaba mucho el filtrado y se tomó la decisión de descartar los datos del 2004. Además, esta decisión estuvo basada en que, al analizar estos datos se veía con claridad que eran de una calidad mucho menor a los de años posteriores, esto fue debido a que fueron los primeros datos publicados por la empresa estando todavía en proceso experimental. De hecho, en el año 2005 se cambió el formato de publicación de los datos, se dejó de publicarlos en archivos de texto con las columnas separadas por el carácter tabulador y se empezó a publicarlos en archivos de valores separados por comas (cvs). También se añadió la columna de identificador de deporte.

Por tanto, una vez descartados los datos del año 2004 se filtraron los datos desde el año 2005 a marzo de 2009 eliminando todas las filas en las que el identificador del deporte fuese distinto de 2 (correspondiente al tenis). Así pues, con este filtro se redujeron los 13,5 GB iniciales a 1,09 GB. Evidentemente en pasos posteriores hubo que reducir todavía más el número de filas de los datos de *Betfair* puesto que había que eliminar las filas correspondientes al tenis femenino y al tenis masculino de dobles. Las columnas utilizadas después de aplicar el filtro son todas a excepción de la columna `SPORTS_ID` que es sobre la que se basó el filtro y al haber sido seleccionado únicamente un valor no aportaba ninguna información. Una vez seleccionados los datos útiles hubo que limpiarlos y depurarlos.

5.2 Limpieza de los datos

En este apartado se tratará de describir como se aumentó la calidad de los datos seleccionados en el apartado anterior. Como se ha señalado anteriormente todavía existían datos del tenis femenino y del tenis dobles masculinos por lo que hubo que eliminar estos subconjuntos de datos.

Para eliminar estas filas del conjunto de datos obtenido en *Betfair* se programaron diversos métodos en Java que aplicaron filtros para encontrar las líneas de apuestas de tenis femenino y dobles masculino.

El primer paso para la eliminación de las filas del tenis femenino primero fue obtener un listado de los nombres de tenistas femeninas, este listado se consiguió de la base de datos de *OnCourt* en la tabla `Players_wta` accediendo al atributo `NAME_P`. A continuación, se recorrieron todas las filas de los datos de *Betfair* buscando coincidencias, tanto en la columna de descripción como en la de selección (`FULL_DESCRIPTION` y `SELECTION`). Al aplicar este algoritmo se comprobó que era compleja su correcta aplicación, pues, en ocasiones daba ciertos problemas y aunque muchas líneas las eliminaba correctamente, otras que quería eliminar eran del tenis masculino, por lo que se decidió cambiar de algoritmo. El siguiente algoritmo aplicado surgió de observar detenidamente los datos de *Betfair*. De esta forma, en la descripción de muchos (la mayoría) de los torneos femeninos aparecían palabras clave como `Ladies, Womens, Fed Cup`[1], `WTA`, etc. Por tanto, simplemente se aplicó un filtro que eliminó las filas en cuya columna `FULL_DESCRIPTION` aparecía cualquiera de las palabras anteriores. De esta forma, se eliminaron la mayoría de los torneos femeninos. Para limpiar el resto de torneos femeninos se buscaron nombres de jugadoras clave, entendiendo por clave, que hubiesen participado en muchos torneos (Sharapova, Henin, Safina, etc) y cuyos nombres no coincidieran con el de ningún jugador masculino, se anotaron los nombres de los torneos que estas jugadoras jugaron y se volvío a filtrar eliminando las apuestas pertenecientes a esos torneos. *Betfair* siempre distinguía los torneos que para masculino y femenino tenían el mismo nombre, para ello introducía palabras clave como las anteriores (por ejemplo *Roland Garros Womens*). Con estos filtros se eliminaron casi todos los partidos del tenis femenino. Los partidos del tenis femenino que todavía no habían sido eliminados se eliminaron de otra forma que se detallará más adelante.

Una vez aplicados los filtros anteriores se redujo el tamaño de los ficheros a 0,88 GB en los que había más de cuatro millones de líneas.

Como se comentó anteriormente también había que eliminar los registros/filas que implicaran a los dobles masculinos. En este caso se pudo aplicar un filtro muy

[1] Torneo exclusivamente femenino, similar a la Copa Davis masculina.

sencillo pues, en las apuestas de este tipo el campo `FULL_DESCRIPTION` casi siempre iba acompañado de la palabra "`Doubles`", por tanto se eliminaron todas las filas en las que se encontrase la palabra "`Doubles`" en dicho campo. Además, otro filtro que se aplicó fue el de buscar en los campos `FULL_DESCRIPTION` o `EVENT` la cadena de caracteres "`*/* v */*`" siendo los asteriscos cualquier cadena de caracteres. Este filtro se aplicó al observar que en los partidos de dobles siempre aparecía una cadena de ese tipo en alguno de los dos campos, ya que ésta era la forma de indicar las dos parejas que jugaban el partido. Un ejemplo de los datos es el siguiente "`Bry/Bry v Dam/Paes`".

También se realizaron otras tareas de limpieza en la base de datos de *OnCourt* ya que contenía datos de dobles masculinos. Para eliminar estas filas se buscó en la tabla `Player_ATP` en la columna `NAME_P` nombres de jugadores en los que se encontrara el carácter '`/`', lo cual indicaba que ese jugador era en realidad una pareja de jugadores. Una vez detectado cada uno de estos nombres se almacenaba su identificador `(ID_P)` y se eliminaban todas las filas de la tabla `Games_ATP` en la que cualquiera de los atributos `ID1_G` o `ID2_G` fuese igual al identificador. También, como es normal, se eliminó la fila con ese identificador en la tabla `Players_ATP` y lo mismo se hizo con las tablas en las que se hiciera referencia a jugadores de dobles como `Rantings_ATP (ID_P_R)`, `Injury_ATP (ID_P_I)`, `Facts_ATP (ID_P_F)`, `Odds_ATP (ID1_O, ID2_O)`, `Seed_ATP (ID_P_S)`, `Stat_ATP (ID1, ID2)` o `Today_ATP (ID1, ID2)`.

Una vez eliminados todos estos datos se consiguió tener en la base de datos de *OnCourt* únicamente datos del tenis masculino individual. En este caso, no como en los datos de *Betfair*, no se encontraron posibles datos no filtrados.

Por lo tanto, se puede decir que después de estos pasos se limpiaron los datos, eso sí no de forma definitiva, puesto que como ya se ha explicado éste es un proceso iterativo. Aún así, esta limpieza permitió seguir adelante con la construcción de los datos.

5.3 Construcción de los datos

En esta tarea se trató de modificar un poco más los datos para posteriormente poder hacer la integración total de los mismos, para ello se crearon atributos derivados o se alteraron valores de atributos ya existentes.

En este paso no se realizaron grandes cambios, ya que,al conjunto de datos anterior sólo se le añadieron dos columnas en la tabla de datos de *Betfair*. Estas columnas recibieron el nombre de `Partido` y `ApuestaJugador`. Hubo que analizar varios atributos de cada línea de datos para saber a qué partido pertenecía y a favor de qué jugador era la apuesta, esto se realizó al integrar los datos. En este punto surgió el problema de encontrarse con líneas de apuestas que no pertenecían a

ningún partido. Por ejemplo, las apuestas a ganador de un torneo, las combinadas o acumuladas (apuestas a favor de varios ganadores en varios partidos distintos), apuestas al progreso de un jugador en un torneo, apuestas a qué jugador de entre dos llegaría más lejos en un torneo, etc. En todas estas filas de apuestas el campo Partido se dejó vacío.

A continuación, se realizaró el paso más importante en la preparación de los datos, que no es otro que la integración y unificación de los mismos en una única base de datos cuyo gestor fue Microsoft Access como se señaló en capítulos anteriores.

5.4 Integración de los datos

En este paso se generó la base de datos final con la que se trabajó a lo largo del proyecto. Una vez empezada la tarea de integración se realizaron cambios pertenecientes a los pasos anteriores lo cual no supuso ningún problema pues ya se advirtió que el proceso de preparación de los datos seguía un ciclo iterativo.

El punto de partida para integrar los datos fue la base de datos de *OnCourt* después de su limpieza. La base de datos ya estaba gestionada por Microsoft Access, que es el gestor que se había elegido al comienzo del proyecto para gestionar la base de datos final, por tanto no hubo que hacer cambios en este sentido.

El siguiente paso fue el de integrar los datos de *Betfair* en esta base de datos. Lo primero fue crear una tabla con el nombre `Betfair`. Esta tabla contenía los siguientes atributos: `Id`, `Partido`, `Id_Evento`, `Cierre_Evento`, `Descripción`, `Previsión_Inicio_Evento`, `Evento`, `Inicio_Evento`, `Id_Apuesta`, `Apuesta`, `Cuota`, `Numero_apuestas`, `Volumen_cruzado`, `Ultima_apuesta`, `Primera_apuesta`, `Ganadora`, `En_juego` y `ApuestaJugador`.

Una vez creada la tabla se introdujeron en ella todos los partidos de tenis que habían quedado después de la selección y limpieza de los datos dejando los campos `Partido` y `ApuestaJugador` vacíos pues fue más adelante cuando fueron rellenados. Después de la inserción de más de cuatro millones de filas la base de datos aumentó su tamaño considerablemente pasando de 33 MB a casi 1 GB.

A continuación, una vez que se introdujeron todos los datos en la base de datos se decidió mejorarla estableciendo relaciones y traduciendo los nombres de tablas y atributos de forma que fuese entendible su significado. Además, se decidió eliminar algunas de las tablas por ser su información irrelevante. Las tablas eliminadas fueron:

- `Today_ATP`: El software *OnCourt* tiene actualizaciones diarias según se celebran los distintos partidos de tenis, en esta tabla se almacenaban los partidos de la fecha actual del sistema los cuales eran irrelevantes para el proyecto.

- `Categories_ATP`: Contenía información de las distintas categorías a las que pertenecían los torneos. Esta información carecía de importancia para el proyecto puesto que todos los partidos de tenistas profesionales (con los que se trabajó a lo largo del proyecto) eran de la misma categoría.

Con la eliminación de estas tablas se acabó de formar la base de datos en cuanto a número de tablas y sus atributos. A continuación, se tradujeron los atributos, siendo necesario en muchas ocasiones interactuar con la interfaz de *OnCourt* para conseguir la traducción, ya que, en las tablas iniciales los nombres de los atributos eran en muchos casos abreviaturas de varias palabras.

A continuación, se describe el contenido de cada una de las tablas y de sus principales atributos.

- **`Jugadores`:** En esta tabla se recogió la información principal de todos los jugadores. Los principales atributos son:
  - `ID`: Identificador único para cada jugador. Atributo numérico.
  - `Nombre`: Nombre y apellidos del jugador. Cadena de caracteres.
  - `Ranking_actual_ATP`: En el tenis masculino hay dos rankings oficiales uno el de la ATP y otro la carrera de campeones cada cual con sus puntuaciones. En este atributo se almacena la última posición del jugador en el ranking ATP. Atributo numérico.
  - `Puntos_XXXX`**:** Son varios atributos que mostrban la puntuación del jugador en el momento actual en cada una de las superficies. Uno de los atributos mostraba el total de puntos en individuales y otro en dobles. Atributos numéricos.
  - `Torneos_XXX`**:** Son varios atributos que contienen el número de torneos en los que ha participado el jugador a lo largo de la temporada actual. Atributos numéricos.
- **`Torneos`:** Recogió información de los distintos torneos que se celebraban en el circuito masculino de tenis. Los principales atributos son:
  - `ID`**:** Identificador único de cada torneo. Las distintas ediciones de los torneos también fueron diferenciadas. Atributo numérico.
  - `Nombre`**:** Nombre del torneo. Cadena de caracteres.
  - `Tipo_pista`**:** Identificador del tipo de la pista del torneo. Hay seis tipos de superficies distintas cada una de las cuales asociada a un identificador único. Atributo tipo fecha.
  - `Fecha`**:** Momento en el que se celebró el torneo. Atributo tipo fecha.

- **`Partidos`:** Almacenó los principales datos de todos los partidos de la base de datos.
    - `ID`**:** Identificador único para cada uno de los partidos almacenados en la base de datos en total más de 57.000. Atributo numérico.
    - `Jugador1`/`Jugador2`**:** Identificadores de cada uno de los jugadores que disputaban el partido. Atributo numérico.
    - `Torneo`**:** Identificador del torneo al que pertenecía el partido. En el mundo del tenis se disputan tantos torneos que no existen los partidos amistosos por lo que todos los partidos disputados pertenecían a algún torneo. Atributo numérico.
    - `Ronda`**:** Identificador de la eliminatoria en la que se jugaba el partido. Atributo numérico.
    - `Resultado`**:** El desenlace final del partido. Una característica observada es que el `Jugador1` siempre es el ganador del partido. Cadena de caracteres.
    - `Fecha`**:** Momento en el que se disputó el partido. Atributo tipo fecha.
- **`Estadísticas_partidos`:** Recogió datos de determinados partidos, en concreto había en la base de datos estadísticas de 16.213 partidos. La mayoría de atributos eran datos propios de un partido de tenis que su propio nombre describía. También estaban los atributos `Jugador1`, `Jugador2`, `Torneo` y `Ronda` que eran redundantes pues se podían encontrar en la tabla `Partidos` accediendo con el `ID` de esta tabla que era el mismo. Su principal atributo era:
    - `ID`**:** Identificador del partido del cual eran las estadísticas almacenadas. Atributo numérico.
- **`Betfair`:** Tabla en la que se insertaron las líneas/filas obtenidas de *Betfair* como detalló anteriormente. Sólo hay tres atributos nuevos en relación a los explicados en el apartado *4.2.2 Datos de Betfair*.
    - `ID`**:** Atributo totalmente nuevo utilizado para distinguir unívocamente cada línea de apuesta. Atributo numérico.
    - `Partido`**:** Relacionaba cada apuesta con el partido al que pertenecía. Si la apuesta no era sobre ningún partido el atributo estaba vacío. Atributo numérico.
    - `ApuestaJugador`**:** Identificador que relacionaba la apuesta con un jugador. Este jugador era por el que se apostaba en esa línea/fila/apuesta. Atributo numérico

- **`Rondas`:** Almacenó las diecisiete distintas rondas que existen en los torneos. Cinco de ellas correspondían a los cinco partidos de las eliminatorias de la Copa Davis y las otras a las distintas eliminatorias del resto de torneos desde la final a las rondas de clasificación. Los atributos eran:
    - `ID`**:** Identificador de la ronda. Atributo numérico con rango desde el 1 hasta el 17.
    - `Nombre`**:** El nombre con el que se conocía a la ronda como por ejemplo “final”. Cadena de caracteres.
- **`Cabezas_de_serie`:** A algunos de los participantes en cada torneo de tenis se les adjudica una “categoría” dentro del torneo que se conoce como cabeza de serie. No todos los torneos tienen los mismos cabezas de serie. Los atributos de esta tabla eran los siguientes:
    - `Jugador`**:** Identificaba el jugador que era cabeza de serie. Atributo numéricra.
    - `Torneo`**:** Identificaba al torneo en el que el anterior jugador es cabeza de serie. Atributo numérico.
    - `Cabeza_de_serie`**:** Identificaba qué tipo de cabeza de serie era. Normalmente iban del uno en adelante, siendo el uno el primer favorito del torneo. Cadena de caracteres.
- **`Pistas`:** Al tenis se juega en diferentes superficies y cada torneo escoge entre ellas siempre que estén dentro del reglamento. Sus dos atributos eran:
    - `ID`**:** Identificaba unívocamente a la pista. Sólo se contemplaban 6 tipos de pistas. Atributo numéricocon rango del 1 hasta el 6.
    - `Nombre_Pista`**:** Nombre de la pista o más concretamente descripción del material de la pista. Cadena de caracteres.
- **`Puntuaciones`:** Cada torneo daba unas puntuaciones según la clasificación del jugador en el mismo, éstas varían entre los distintos torneos y sus distintas ediciones por lo que se almacenan en esta tabla. Los atributos eran todos numéricos y correspondían a los puntos que se daban en cada una de las rondas.
    - `ID`**:** Identificador de la puntuación en cuestión. A este identificador se hacía referencia desde la tabla `Torneo` para indicar que un torneo puntuaba según la puntuación indicada en esta tabla. Atributo numérico.
    - `Ganador`**:** Puntuación del ganador del torneo. Atributo numérico.

  - `Finalista`**:** Puntuación del finalista no ganador del torneo. Atributo numérico.
  - `Primera_ronda_clasificación`**:** Puntuación de los clasificados a la primera ronda clasificatoria del torneo. Atributo numérico.
  - Un atributo para cada una de las rondas que había en los distintos torneos.

- **`Resumen_datos`:** En esta tabla se resumían las estadísticas de todos los jugadores que habían jugado algún partido en el año indicado en el atributo `Año`. Sus principales atributos eran:
  - `Jugador`**:** Identificaba al jugador al que pertenecían los datos del resto de atributos. Atributo numérico.
  - `Año`**:** Año en el que consiguieron las estadísticas. Atributo numérico.

- **`Lesiones`:** Cuando un jugador abandonaba un partido por lesión con la correspondiente derrota esto quedaba reflejado en esta tabla. Los atributos de eran los siguientes:
  - `ID`**:** Identificador único de la lesión de un jugador en un momento dado. Atributo numérico.
  - `Jugador`**:** Identificaba al jugador que se produjo la lesión. Atributo numérico.
  - `Fecha`**:** Indicaba el momento en el que se produjo la lesión. Es de tipo fecha.
  - `Descripción`**:** Describía el tipo de lesión, por ejemplo hombro, espalda, etc. Cadena de caracteres.

- **`Ratings`:** En esta tabla se almacenaban los puntos y la posición de cada uno de los jugadores en cada una de las semanas del año. Los atributos eran los siguientes.
  - `Fecha`**:** Indicaba en qué momento el jugador tenía los puntos y la posición indicadas en los siguientes atributos. Atributo tipo fecha.
  - `Jugador`**:** Identificaba al jugador con esa puntuación y posición en el momento indicado en el atributo anterior. Atributo numérico.
  - `Puntos`**:** Almacenaba los puntos que tenía el jugador en la fecha indicada en el atributo `Fecha`. Atributo numérico.
  - `Posición`**:** Almacenaba la posición que ocupaba el jugador en el ranking ATP en el instante indicado en `Fecha`. Atributo numérico.

- **`Links`:** Almacenaba enlaces interesantes con información del mundo del tenis. Se conservó esta tabla porque podría haber sido de ayuda en ciertos momentos para realizar algunas consultas. Su atributo principal era:
    - `Link`: dirección web con información tenística. Cadena de caracteres.
- **`Urls`:** Tabla casi idéntica a la anterior y con información semejante. Su atributo principal también era:
    - `Link`: dirección web con información tenística. Cadena de caracteres.

Una vez descritas todas las tablas del proyecto y sus principales atributos falta describir como fueron calculados los atributos derivados de `Partido` y `ApuestaJugador` en la tabla `Betfair`. Este fue uno de los pasos que más complicaciones produjo, pues, *Betfair* cambiaba el formato de descripción de los eventos constantemente y no había una fórmula única para el cálculo de los atributos.

El primer punto de partida para rellenar los dos atributos era saber si la línea de datos de la tabla *Betfair* era de apuestas a un partido o a otra cosa. Este punto no fue demasiado complicado, pues, en todas las líneas de apuestas a partidos en los campos `Descripción` o `Evento` siempre aparecía la cadena de caracteres `" v "` con los espacios incluidos. Para los aficionados al mundo del deporte esta cadena de caracteres es muy conocida y viene de la palabra inglesa *versus* cuyo significado es "frente a". Por tanto, una vez encontrada esta cadena en cualquiera de los dos atributos indicaba que antes y después de ella se encontraban los nombres de los dos jugadores que participaban en el partido. Analizando los datos se observó que si la cadena era encontrada en el campo `Evento` sólo se encontraba esta cadena junto con los dos nombres de los jugadores que disputaban el partido, sin embargo, si la cadena se encontraba en el campo `Descripcion` antes y después de los nombres de los jugadores podía haber otros datos, pero a su vez, estos datos estaban separados por el carácter `'/'`. Por tanto, se programó una aplicación en Java que consiguiera determinar los jugadores que disputaban el partido mediante consultas a la tabla `Jugadores` de la base de datos. Como se ha descrito antes, esto no fue tan sencillo, puesto que, *Betfair* no llamaba a un jugador de la misma forma que estaba registrado en la tabla `Jugadores`, además a lo largo de los años también variaba su forma de nombrarlos, por ejemplo el tenista Guillermo García López a veces es mencionado como García López, otras García-López, García L. o G. López.

Ante esta situación para que las consultas realizadas tuvieran éxito se sustituyeron todos los caracteres especiales como `'.'`, `'-'` o `'/'` por el carácter `'%'` que en consultas en lenguaje SQL representa cualquier cadena de caracteres. Una vez se había detectado por lo menos uno de los dos jugadores que participaban en el partido, se consultaban sus partidos en la fecha indicada por el campo `Cierre_Evento` y en los días anterior y posterior. Esto fue realizado de esta forma para no tener problemas con posibles variaciones en la fecha entre las dos fuentes de datos, *Betfair* y *OnCourt*. Como máximo por tanto se obtenían tres posibles partidos

(los tenistas no juegan más de un partido de individuales al día) a los que la fila tenía que corresponder. Para saber cuál de estos partidos era el correcto simplemente había que coger una subcadena del nombre del otro jugador que lo disputaba y ver con cuál de los rivales de los partidos coincidía. Si por casualidad la subcadena coincidía con más de uno se ampliaba en un carácter hasta que la coincidencia fuera única. Una vez se averiguaba el partido, se introducía su identificador en la columna `Partido` de la tabla `Betfair` y de esta forma se relacionaba la fila de `Betfair` con el partido al que correspondía esa fila/apuesta.

Para completar la base de datos faltaba insertar en el campo `ApuestaJugador` de la tabla `Betfair` el jugador por el que se apostaba a favor en esa fila. Al tener ya el partido, este punto fue mucho más fácil, ya que, con el identificador del partido se podía acceder a los dos jugadores que intervenían en él y comparar sus nombres con el texto que se encontraba en el campo `Apuesta` de la tabla `Betfair`, el que coincidiera era el jugador por el que se apostaba en esa fila. Ya sólo quedaba insertar su identificador en el campo `ApuestaJugador` de la tabla `Betfair`. Una vez realizado este paso para los más de cuatro millones de filas de la tabla `Betfair` casi estaría terminada la base de datos.

Por último, faltaba quizás la tarea más tediosa de todas que era comprobar los errores al insertar en estas filas y corregirlos manualmente. Cuando se detectaba en los campos `Descripción` o `Evento` que la fila contenía un partido porque se encontraba la cadena `" v "` pero no se encontraba el partido en la tabla `Partidos`, ya fuera por no encontrar los jugadores o por no coincidir la fecha, se almacenaba en el campo `Partido` un -1. Se filtraron mediante Access los resultados para que sólo fueran mostrados los que tuvieran el valor -1 en el campo `Partido` y manualmente se corrigieron. La mayoría de estos errores venían de competiciones en las que se compite por equipos como la Copa Davis o la Copa Hopman. En estos casos, como jugador se ponía a veces el nombre del equipo, por ejemplo “Spain” y como es normal el algoritmo no detectaba al jugador “Spain” en la tabla `Jugadores`. Una vez corregidos todos los fallos, se había terminado con la elaboración de la base datos, la cual se muestra a continuación a modo de diagrama en la Figura 6. Al ser demasiado grande únicamente se muestran las distintas tablas con sus correspondientes claves principales y claves foráneas. Para consultar todos los atributos de todas las tablas de la base de datos ver el *Anexo A: Tablas de la base de datos*.

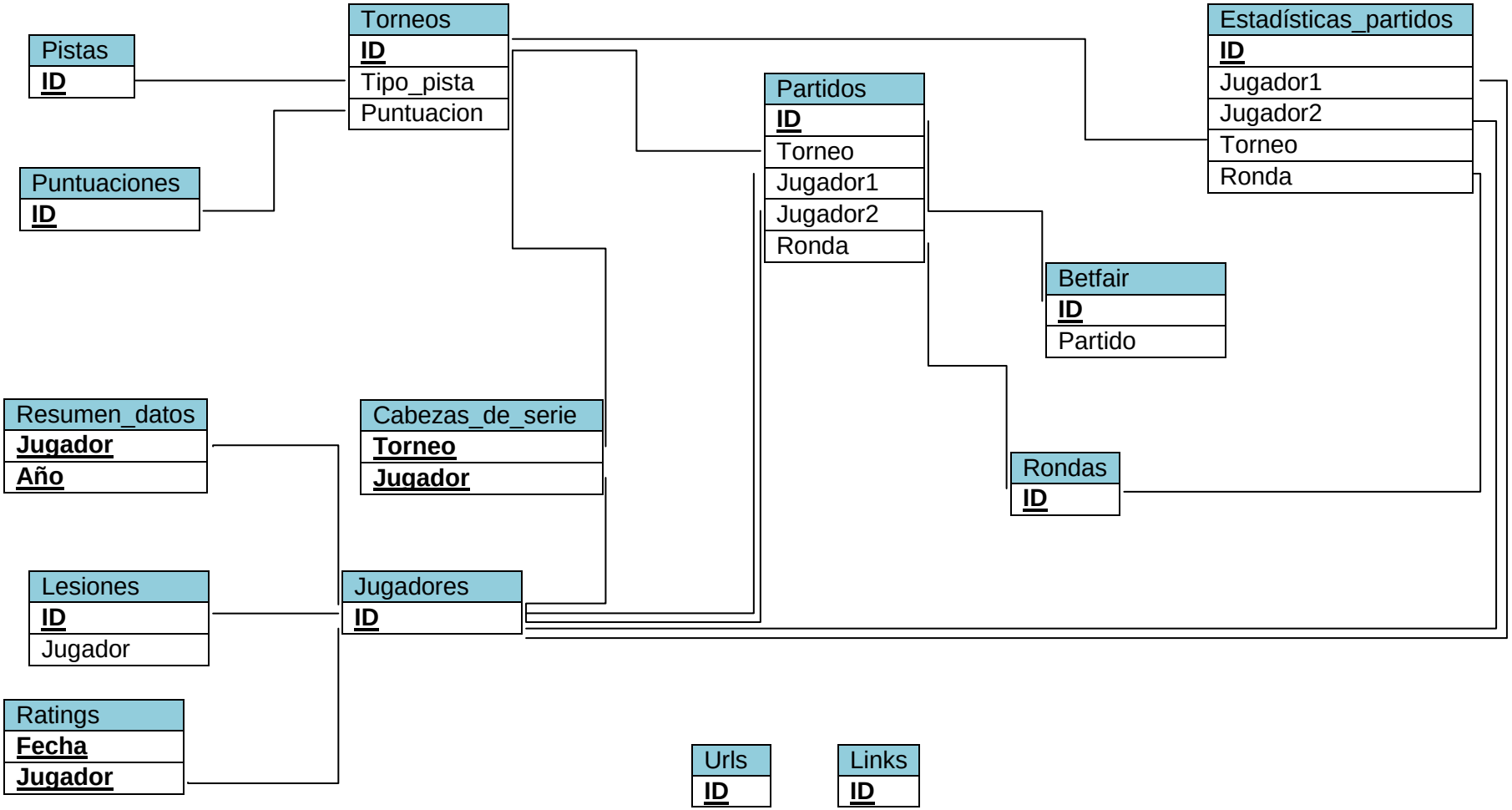


Figura 6: Diseño final de la base de datos

5.5 Formato de los datos

En este proyecto no procede el formato de los datos de la forma que viene definido en la metodología CRISP-DM. En esta metodología se especifica que en este apartado se realizan modificaciones sintácticas en los datos que no cambien su significado pero que sí son requeridas por la herramienta de modelado. En este caso, la base de datos fue creada para poder formar los ficheros de datos para la herramienta de modelado de una forma más simple y eficiente.

Como se especificó en apartados anteriores la herramienta de modelado elegida para la realización del proyecto fue Weka. Aunque Weka acepta en teoría ficheros csv, como los obtenidos inicialmente de *Betfair*, e incluso soporta consultas a bases de datos, las que se podrían hacer a la base de datos inicial de *OnCourt*, surgen muchos problemas para cargar datos en los dos formatos. Por tanto se tomó la decisión de crear ficheros con formato ARFF[1] mediante una aplicación en Java que haciese las consultas deseadas a la base de datos final especificada en el apartado anterior. De esta forma, se generaron los distintos ficheros con los que se trabajó en Weka.

La estructura de un fichero con formato ARFF es muy sencilla. Los ficheros con formato ARFF se dividen en tres partes: `@relation, @attribute y @data`

- `@relation <nombre>`: Todo fichero ARFF debe comenzar con esta declaración en su primera línea (no se pueden dejar líneas en blanco al principio). `<nombre>` será una cadena de caracteres y si contiene espacios se pondrá entre comillas.

- `@attribute <nombre> <tipo_de_datos>`: En esta sección se incluye una línea por cada atributo (o columna) que se vaya a incluir en el conjunto de datos, indicando su nombre y el tipo de dato. Con `<nombre>` se indicará el nombre del atributo, que debe comenzar por una letra y si contiene espacios tendrá que estar entrecomillado. Con `<tipo_de_datos>` se indicará el tipo de dato para este atributo (o columna) que puede ser:

    - `Numeric`: Numérico.

    - `String`: Texto.

[1] Formato propio de los ficheros con los que trabaja Weka.

  - `date [<formato_de_fecha>]`: Fecha. En `< formato_de_fecha >` se indicará el formato de la fecha, que será del tipo `"yyyy-MM-dd'T'HH:mm:ss"`.

  - `<nominal-specification>`. Estos son tipos de datos definidos por el usuario y que pueden tomar una serie de valores que se indican entre llaves y separados por comas como por ejemplo: `"@attribute tiempo {soleado, nuboso, lluvias}"`.

- `@data`: En esta sección son incluidos los datos propiamente dichos. Cada columna es separada por comas y todas filas deben tener el mismo número de columnas, número que coincide con el de declaraciones `@attribute` añadidas en la sección anterior. Si no se dispone de algún dato, se escribe un signo de interrogación `(?)` en su lugar. El separador de decimales tiene que ser obligatoriamente el punto y las cadenas de tipo `string` tienen que estar entre comillas simples.

Una vez conocida la estructura de los ficheros ARFF de Weka sólo hubo que programar pequeñas aplicaciones que generaran los distintos ficheros necesarios para los distintos experimentos diseñados. Estos ficheros no son descritos en este apartado, pues, formarán parte del siguiente capítulo en el que se detallarán los distintos experimentos realizados.

La mayor dificultad encontrada en la generación de estos ficheros fue el simple hecho de tener que calcular atributos derivados como por ejemplo las cuotas de cada evento. Cada evento tenía numerosos registros con distintas cuotas a favor de uno y otro jugador. El cálculo más complicado fue el de la cuota media ya que había que tener en cuenta varios atributos de la tabla `Betfair`.

Sin embargo, no fue la complejidad lo que más retraso esta fase sino el tiempo que tardaba la aplicación, una vez ser ponía en marcha, en generar cada uno de los ficheros puesto que, al ser tantos registros (más de cuatro millones) se tenían que hacer muchas consultas a la base de datos y normalmente el ordenador estaba días funcionando hasta acabar la generación de cada uno de los ficheros ARFF. Este problema también se tuvo al crear la base de datos final y tener que hacer todas las inserciones en la tabla `Betfair`.

Capítulo 6: MODELADO

En este capítulo, se describirán las distintas técnicas de modelado seleccionadas y los parámetros aplicados en cada una de ellas. Se aplicaron tareas de preparación de los datos para generar nuevos ficheros de datos a los que aplicar las técnicas elegidas. Una vez aplicadas las técnicas de modelado, a los distintos conjuntos de datos, los distintos resultados obtenidos fueron recopilados y comparados.

6.1 Selección de las técnicas de modelado

En esta fase fueron seleccionadas varias técnicas de modelado que posteriormente fueron aplicadas, parcialmente o en su totalidad, a los distintos conjuntos de datos para llevar a cabo los experimentos diseñados.

Al comienzo del proyecto se realizó un planteamiento inicial de la situación para definir de una forma más clara el problema. Para ello se enumeraron las distintas tareas tratadas por la minería de datos analizando cuál de ellas era la más adecuada para el problema planteado en el proyecto.

Las principales tareas en problemas de minería de datos pueden ser divididas en dos grupos: predictivas y descriptivas. En este caso se tomó la decisión de contemplar únicamente las tareas predictivas, ya que, como se describió en apartados anteriores uno de los objetivos principales del proyecto era el de calcular la rentabilidad, conseguida al apostar, utilizando los modelos de predicción creados.

Las principales tareas predictivas son las siguientes:

- **Clasificación:** En los problemas de clasificación normalmente se dispone de una base de datos compuesta por un número N de ejemplos o

instancias que están descritos por un número `P` de atributos y que pertenecen a una clase. En estos problemas se trata de aprender la forma de distinguir los ejemplos de las distintas clases. El problema planteado en este proyecto puede ser tratado como un problema de clasificación en el que la clase es el jugador que haya ganado el partido. La forma de enunciar el problema podría ser la siguiente: Se dispone de una base de datos amplísima de partidos con datos de cada uno de los jugadores. Los partidos son jugados por dos jugadores a los que se nombraría como `Jugador1` y `Jugador2`. La clase sería el ganador del partido, por tanto, sólo admite dos posibles valores o `Jugador1` o `Jugador2`. De cada uno de los jugadores se podrían disponer de un gran número de datos que servirían para predecir el ganador, es decir, la clase (`Jugador1` o `Jugador2`). Por tanto, el problema que se afronta en este proyecto puede ser tratado como un problema de clasificación.

- **Regresión:** Los problemas de regresión son similares a los de clasificación, la diferencia radica en que en estos problemas el valor que se trata de predecir es un valor continuo. Este proyecto podría ser tratado en algunos puntos con tareas de regresión, por ejemplo, se podrían usar tareas de regresión para calcular las cuotas de un partido. Antes del comienzo de un partido las cuotas a favor de uno y otro jugador varían según se acerca el comienzo del partido. Con tareas de este tipo se podrían tratar de predecir cuales serán los valores máximos y mínimos de las cuotas.

- **Análisis de series temporales:** Una serie temporal es una sucesión de observaciones de una variable tomadas en varios instantes de tiempo. El objetivo de estas tareas es estudiar los cambios en esa variable con respeto al tiempo y predecir sus valores futuros. A menudo, se representa la serie en un gráfico temporal, con el valor de la serie en el eje de ordenadas y los tiempos en el eje de abscisas. En el ámbito del proyecto éste sería uno de los estudios más interesantes siendo la variable a observar la cuota a favor de un jugador a lo largo del tiempo. Predecir el valor futuro de la cuota en este caso sería similar a predecir el resultado del evento, ya que, la cuota es inversamente proporcional a la probabilidad del resultado. Sin embargo, este estudio no fue posible realizarlo pues los datos de *Betfair* no ofrecen los movimientos de fluctuación de las cuotas a lo largo del tiempo, sólo ofrecen para cada valor de una cuota a la que se apostó, el volumen apostado, el número total de apuestas, el instante en el que se produjo la primera apuesta a esa cuota y el último instante en el que se produjo la última apuesta a esa cuota. Para que estos datos estuvieran completos harían falta todos los instantes en los que se apostó a cada cuota, no sólo el primero y el último.

Resumidas las principales tareas predictivas hay que volver a señalar que la tarea elegida para llevar a cabo el desarrollo del proyecto fue la tarea de clasificación por ser la más adecuada para el dominio del problema.

6.1.1 Clasificación

Una vez seleccionada la tarea principal, se debía elegir qué algoritmos de clasificación serían utilizados para llevar a cabo el proyecto. Los algoritmos fueron elegidos en base a las anotaciones aportadas por un experto en el dominio, el cual, ya había participado en otros proyectos similares.

Los algoritmos de clasificación utilizados a lo largo de la experimentación fueron los siguientes:

- **ZeroR:** Es el algoritmo de clasificación más sencillo que existe. Todas las instancias son clasificadas como la clase mayoritaria. Es usado como caso base para hacer comparaciones con otros algoritmos, cualquiera de los otros algoritmos como mínimo debería de igualar su resultado.

- **OneR:** Este clasificador construye un clasificador consistente en usar una única variable en el antecedente, es decir, se genera una regla que clasifica a un objeto en base a un solo atributo. Se generan todas las reglas del tipo "`Si variable = valor Entonces clase = categoría`" para una única variable. Este algoritmo se fundamenta en la tesis de que reglas de clasificación muy sencillas trabajan bien en la mayoría de las bases de datos empleadas. También suele usarse como algoritmo base para realizar comparaciones (Holte, 1993).

- **DecisionTable:** A partir de los datos de entrenamiento construye una tabla formada por un subconjunto (llamado esquema) de los atributos y una selección de las instancias de entrenamiento. Para clasificar una nueva instancia el algoritmo busca en la tabla todos los ejemplos que concuerdan, teniendo en cuenta sólo los atributos que forman el esquema. Si no se encuentra ningún ejemplo que concuerde, el algoritmo devuelve la clase más cercana de la tabla; en otro caso, devuelve la clase mayoritaria del conjunto de ejemplos que concordaron (Kohavi, 1995).

- **NaiveBayes:** El razonamiento bayesiano brinda un enfoque probabilístico a la inferencia. Está basado en el supuesto de que los valores de interés están gobernados por distribuciones de probabilidad y que pueden tomarse decisiones óptimas razonando sobre estas probabilidades junto con los datos observados. El algoritmo NaiveBayes está entre los más prácticos y efectivos métodos para muchos problemas de aprendizaje. Este algoritmo se basa en la hipótesis de que las variables que describen a las instancias son estadísticamente independientes. En la mayoría de las ocasiones esto no es verdad, sin embargo, con frecuencia, esta simplificación del problema arroja resultados con una buena aproximación. A partir del conjunto de entrenamiento se calcula la probabilidad a priori de que una instancia cualquiera pertenezca a una clase, también se calcula la probabilidad condicional de que un atributo tome un valor si la instancia pertenece a una determinada clase. A

continuación, con estos datos se puede calcular, utilizando la fórmula de Bayes y asumiendo independencia entre las variables, la probabilidad de que una instancia pertenezca a una clase si sus atributos toman determinados valores. La clasificación de la instancia dada será la que haga máxima esta probabilidad (Hart, 1973).

- **C4.5:** Forma parte de los árboles de decisión. Los árboles de decisión se construyen comenzando por la raíz hasta las hojas. Primeramente se escoge un atributo para discriminar y se produce un sub-nodo por cada valor del atributo. Si todos los ejemplos con un valor particular de atributo tienen la misma clase, el nodo se convierte en hoja, de otra forma se escoge otro atributo para seguir discriminando entre las clases. El árbol está completo cuando todos los ejemplos son representados por un nodo hoja. Para determinar qué atributo se ramifica en cada nivel se calcula la información ganada al discriminar con cada atributo y se usa aquel que maximice la ganancia. Un árbol aprendido puede representarse también como un conjunto de reglas `"si-entonces"`, más fáciles de entender para un usuario. Este algoritmo genera árboles de decisión para atributos discretos y continuos, utiliza la razón de ganancia para seleccionar el atributo de cada nodo y aplica estrategias de poda para reducir el ruido de los datos de entrenamiento (Quinlan, 1993).

- **IB1 (Vecino más cercano):** Este algoritmo pertenece a la familia de los algoritmos basados en instancias u holgazanes. Estos algoritmos clasifican una instancia comparándola con una base de datos de ejemplos pre-clasificados. La principal suposición que se hace es que instancias similares tendrán clasificaciones similares. Se les llama "holgazanes" porque realizan poco trabajo en la etapa de aprendizaje, en los casos más simples tan solo se almacenan los ejemplos en memoria, transfiriendo el esfuerzo al momento de clasificar una nueva instancia, cuando el sistema debe decidir cuáles de los ejemplos memorizados debe utilizar para hacer la clasificación. Este algoritmo viene del algoritmo general conocido como IBk (k vecinos más cercanos), en este caso `k=1`, es decir, el vecino más cercano. Cuando se le proporciona una nueva instancia, este algoritmo busca, entre las que se utilizaron durante el entrenamiento, la instancia más parecida, y la clasifica como tal. Utiliza la distancia Euclidea para medir similitud entre instancias.

- **Meta clasificadores:** Además de buscar el algoritmo que individualmente proporciona mejores resultados para un problema determinado, existe una vía alternativa para mejorar aún más la precisión: agrupando los clasificadores en conjuntos (Dietterich, 1997), también llamados meta clasificadores. Un conjunto es un grupo de clasificadores cuyas predicciones individuales son combinadas de alguna manera (típicamente mediante el voto) para clasificar nuevos ejemplos. Esta es una de las áreas más activas de investigación en aprendizaje supervisado ya que los conjuntos son mucho más precisos que los clasificadores individuales que los componen (Matjaž Gams, 1994). La mayoría de las

investigaciones sobre conjuntos de clasificadores se concentran en la generación de conjuntos con un solo algoritmo de aprendizaje, llamados modelos homogéneos (Dietterich, 2000). Diferentes clasificadores son generados manipulando el conjunto de entrenamiento (como hacen *boosting* o *bagging*), manipulando las atributos de entrada, manipulando la salida o inyectando ruido en el algoritmo de aprendizaje. Los clasificadores generados son generalmente combinados por votación ponderada o mayoritaria. Otro enfoque es generar los clasificadores aplicando diferentes algoritmos de aprendizaje a un solo conjunto de datos, con representación heterogénea de modelos. En el presente estudio se emplearon los meta clasificadores *bagging* y *boosting*, que están entre los más utilizados.

  - **Bagging**: La idea básica que subyace en *bagging* es que el error del modelo construido por un clasificador se debe en parte a la selección de un conjunto de entrenamiento específico. Por tanto, si se crean varios conjuntos de datos tomando muestras con reemplazo y se crean sendos clasificadores, se reducirá el componente de varianza del error de salida (Peter Buhlmann, 2002).

  - **Boosting*:*** Se reajustan repetidamente los pesos de los ejemplos de entrenamiento, concentrándose en los ejemplos "dificultosos" de muestras anteriores. A los ejemplos que son mal clasificados se les asigna mayor peso en la próxima iteración, por ejemplo, las instancias que están próximas a la frontera de decisión son a menudo más difíciles de clasificar, y por tanto, adquieren mayor peso tras pocas iteraciones (Meir R, 2003).

- **REPTree**: Construye un árbol de decisión usando la ganancia de información y realiza una poda de error reducido. Solamente ordena una vez los valores de los atributos numéricos. Los valores ausentes se manejan dividiendo las instancias correspondientes en segmentos (Witten I., 2000).

- **ConjuctiveRule:** Es un algoritmo que genera un clasificador simple de reglas conjuntivas.

6.1.2 Selección de atributos

Al disponer de una gran cantidad de atributos, el proceso de clasificación tenía un coste computacional alto, ya que, algunos ficheros de datos estaban formados por más de 57.000 ejemplos con más de 170 atributos cada uno. Hubo que tomar decisiones que solucionaran este problema, pues, se tardaba días en obtener la solución de cada uno de los experimentos, y además, los resultados no eran demasiado buenos. De esta forma, se llegó a la conclusión de que para reducir el

tamaño de los ficheros de entrenamiento y test, y además, como posible solución a los malos resultados obtenidos, había que aplicar la tarea de selección de atributos.

La selección de atributos, como su nombre indica, selecciona un subconjunto de los atributos originales. Las ventajas esperadas de este proceso son:

- Mejorar el desempeño predictivo.
- Construir modelos más eficientemente.
- Mejorar el entendimiento de los modelos generados.

Aunque en la práctica muchos atributos pueden traer un mayor poder discriminativo con una cantidad limitada de datos, una cantidad excesiva de atributos retrasa significativamente el proceso de aprendizaje y frecuentemente produce sobre-ajustes.

El proceso de selección de atributos trata de seleccionar el subconjunto más pequeño de atributos tal que no se afecte significativamente el porcentaje de acierto en la clasificación. Un atributo se considera relevante si no es irrelevante o redundante. Un atributo es irrelevante si no afecta de ninguna forma el concepto meta y es redundante si no añade nada nuevo al concepto meta.

El proceso de selección de atributos involucra 4 pasos:

- Generación de candidatos (subconjuntos), lo cual involucra una estrategia de búsqueda.
- Evaluación de candidatos (subconjuntos), que requiere un criterio de evaluación.
- Criterio de parada: Puede darse por la estrategia de búsqueda, el número de iteraciones realizadas, el número de atributos seleccionados, el que no se mejore el criterio de evaluación al añadir (quitar) otro atributo, que el error de clasificación esté por debajo de un umbral, etc.
- Validación de resultados: Si se sabe de entrada cuáles son los atributos relevantes, se puede comparar el resultado del algoritmo con esos atributos conocidos. Como normalmente no se sabe, se puede comparar el error en la clasificación con y sin la selección de atributos.

Se utilizaron tres algoritmos evaluadores de subconjuntos de atributos de los disponibles en Weka. El primero de ellos está clasificado como filtro y se aplicó primero con el método `RankSearch`, que ordena los atributos y crea un ranking de subconjuntos, después se aplicó con el método `GeneticSearch`, que es un algoritmo genético de búsqueda. Los dos algoritmos restantes empleados son evaluadores de atributos individuales y cada uno se aplicó unido al método `Ranker`, que devuelve una lista ordenada de los atributos según su calidad:

- **`CfsSubsetEval`**: Evalúa un subconjunto de atributos considerando la habilidad predictiva individual de cada variable, así como el grado de

redundancia entre ellas. Se prefieren los subconjuntos de atributos que estén altamente correlacionados con la clase y tengan baja intercorrelación (Hall M. A., 1998).

- **`ChiSquaredAttributeEval`**: Calcula el valor estadístico Chi-cuadrado de cada atributo con respecto a la clase y así obtiene el nivel de correlación entre la clase y cada atributo.

- **`InfoGainAttributeEval`**: Evalúa los atributos midiendo la ganancia de información de cada uno con respecto a la clase. Anteriormente discretiza los atributos numéricos (Lorenzo, 2002).

6.2 Generación del diseño del experimento

En este apartado se describirá la planificación de los experimentos que se llevaron a cabo. Se indicarán los conjuntos de datos utilizados para entrenamiento y para test además de cómo se evaluaron los resultados. Se realizaron a lo largo del proyecto distintos experimentos con distintos subconjuntos de datos. A continuación, se enumeran y describen de forma general los experimentos realizados.

- **Estadísticas completas**: En este experimento el conjunto de datos estaba formado por ejemplos compuestos por distintas estadísticas de los jugadores que disputaban un partido y algunos datos del partido en cuestión. En total eran 152 atributos y 57.400 partidos/líneas/ejemplos. El objetivo era predecir el ganador del partido aumentando el porcentaje de acierto lo máximo posible. Se aplicaron algunos de los algoritmos anteriormente descritos, cuyos resultados se detallan en el siguiente apartado. Se realizó validación cruzada de diez subconjuntos.

- **Apuestas Pre Inicio:** El conjunto de datos estaba formado por ejemplos con datos de apuestas de antes del comienzo de los partidos. En concreto, en este caso cada ejemplo únicamente tenía cinco atributos (`Cuota, Numero_Apuestas, Volumen, EnJuego y Ganada`). En total se disponía de 806.908 ejemplos siendo el objetivo predecir si la apuesta resultaba ganadora intentando aumentar el porcentaje de acierto en la predicción. Los algoritmos aplicados serán detallados en el siguiente apartado. El experimento se llevó a cabo con validación cruzada de diez subconjuntos.

- **Apuestas En Vivo:** El conjunto de datos estaba formado por ejemplos con datos de apuestas en vivo, es decir, una vez habían comenzado los partidos. Cada ejemplo únicamente tenía cuatro atributos (`Cuota, Numero_Apuestas, Volumen y Ganada`). En total se disponía de 2.340.325 ejemplos siendo el objetivo predecir si la apuesta resultaba

ganadora intentando aumentar el porcentaje de acierto en la predicción. Los algoritmos aplicados serán detallados en el siguiente apartado. En este experimento no se aplicó validación cruzada porque el software tiene limitaciones al trabajar con archivos con tantas instancias y no funcionaba correctamente. Por tanto, se dividieron los ejemplos en dos ficheros, uno de entrenamiento y otro de test, el de entrenamiento constaba de 1.545.755 ejemplos y el de test de 794.564. Los ficheros se formaron con ejemplos tomados aleatoriamente, formando el fichero de entrenamiento con el 66% de los ejemplos y el de test con el 33% restante.

- **Estadísticas completas con selección de atributos:** En este experimento se generaron cuatro ficheros de datos nuevos, cada uno de estos ficheros estaba formado con los atributos seleccionados por cuatro algoritmos de selección de atributos aplicados al fichero formado para el experimento con las estadísticas completas. Una vez creados los ficheros se aplicaron distintos algoritmos de clasificación y se compararon los resultados entre estos y con los obtenidos en el experimento de estadísticas completas cuyo objetivo era similar al de este experimento, es decir, predecir correctamente el mayor número de partidos. Se aplicó siempre validación cruzada de diez subconjuntos.

- **Estadísticas separadas sin cuotas:** En este experimento se generó un fichero de datos con todos los partidos de los que se tenían datos de apuestas en la tabla `Betfair`. Cada ejemplo tenía como atributos estadísticas de los jugadores que disputaban un partido y algunos datos del partido. En total, cada ejemplo estaba compuesto por 153 atributos y se disponía de 12.673 ejemplos. Se aplicaron distintos algoritmos de clasificación y se realizó validación cruzada de diez subconjuntos.

- **Estadísticas separadas con cuotas:** En este experimento se generó un fichero de datos con todos los partidos de los que se tenían datos de apuestas en la tabla `Betfair`. Cada ejemplo tenía como atributos estadísticas de los jugadores que disputaban un partido, algunos datos del partido y datos de las apuestas a favor de cada uno de los jugadores. En total cada ejemplo estaba compuesto por 161 atributos y se disponía de 12.673 ejemplos. Se aplicaron distintos algoritmos de clasificación y se realizó validación cruzada de diez subconjuntos.

- **Estadísticas enfrentadas sin cuotas:** En este experimento se generó un fichero de datos con todos los partidos de los que se tenían datos de apuestas en la tabla `Betfair`. Cada ejemplo tenía como atributos estadísticas de los jugadores que disputaban un partido enfrentadas, es decir, en todas las estadísticas numéricas se hizo la resta de los datos del jugador 1 menos los del jugador 2. Por ejemplo,si el jugador 1 tenía 530 puntos en el ranking ATP, en el momento del partido, y el 2 430, el valor del dato en el fichero sería de -100. También había atributos que daban información del partido y en total cada ejemplo estaba compuesto por 71 atributos. Se disponía en total de 12.673 ejemplos. Se aplicaron

distintos algoritmos de clasificación y se realizó validación cruzada de diez subconjuntos.

- **Estadísticas enfrentadas con cuotas:** En este experimento se generó un fichero de datos con todos los partidos de los que se tenían datos de apuestas en la tabla `Betfair`. Cada ejemplo tenía como atributos estadísticas de los jugadores que disputaban un partido enfrentadas. También había atributos que daban información del partido y datos de las apuestas a favor de cada uno de los jugadores. En total, cada ejemplo estaba compuesto por 79 atributos y se disponía de 12.673 ejemplos. Se aplicaron distintos algoritmos de clasificación y se realizó validación cruzada de diez subconjuntos.

- **Estadísticas separadas sin cuotas y selección de atributos:** Fue idéntico al experimento "Estadísticas separadas sin cuotas" con la salvedad de que cada ejemplo sólo disponía de los 20 mejores atributos seleccionados mediante el algoritmo de selección de atributos `InfoGainAttributeEval` con el método `Ranker` más el atributo de la clase.

- **Estadísticas separadas con cuotas y selección de atributos:** Fue idéntico al experimento "Estadísticas separadas con cuotas" con la salvedad de que cada ejemplo sólo disponía de los 20 mejores atributos seleccionados mediante el algoritmo de selección de atributos `InfoGainAttributeEval` con el método `Ranker` más los ocho atributos de las cuotas a favor de los dos jugadores y el atributo de la clase.

- **Estadísticas enfrentadas sin cuotas y selección de atributos:** Fue idéntico al experimento "Estadísticas enfrentadas sin cuotas" con la salvedad de que cada ejemplo sólo disponía de los 20 mejores atributos seleccionados mediante el algoritmo de selección de atributos `InfoGainAttributeEval` con el método `Ranker` y el atributo de la clase.

- **Estadísticas enfrentadas con cuotas y selección de atributos:** Fue idéntico al experimento "Estadísticas enfrentadas con cuotas" con la salvedad de que cada ejemplo sólo disponía de los 20 mejores atributos seleccionados mediante el algoritmo de selección de atributos `InfoGainAttributeEval` con el método `Ranker` más los ocho atributos de las cuotas a favor de los dos jugadores y el atributo de la clase.

- **Simulación final con estadísticas separadas sin cuotas:** En este experimento se generaron 14 ficheros de datos para el entrenamiento con sus correspondientes 14 ficheros de test. Los ficheros de datos estaban formados por todos los partidos de los que se tenían datos de apuestas en la tabla `Betfair` hasta una fecha determinada, empezando por enero de 2008 y hasta marzo de 2009. Los ficheros de entrenamiento

tenían los partidos hasta empezar un mes, por ejemplo, el fichero "Partidos_hasta_mayo_2008", y los de test los partidos del siguiente mes, por ejemplo, el fichero "Partidos_de_mayo_2008". Cada ejemplo de los ficheros tenía como atributos estadísticas de los jugadores que disputaban el partido y algunos datos del partido. El algoritmo de clasificación aplicado fue el que de media tuvo mejores resultados en los experimentos anteriores. Como se ha mencionado anteriormente, los test de las pruebas se hicieron a través de los ficheros de test correspondientes. La valoración de los resultados varió con respecto a los experimentos anteriores, se dieron datos del porcentaje de acierto de los partidos pero además, se detallaron resultados de varias simulaciones de inversión en el mundo de las apuestas, detallando posibles ganancias o pérdidas según la cantidad apostada o la cuota a la que se apostase.

- **Simulación final con estadísticas separadas con cuotas:** Este experimento fue muy similar al anterior y únicamente varió respecto a éste en que los ficheros de datos (entrenamiento y test), además de tener como atributos estadísticas de los jugadores que disputaban un partido y algunos datos del partido, fueron añadidas las cuotas a favor de cada uno de los jugadores que disputaban cada partido. El algoritmo de clasificación aplicado también fue el que de media tuvo mejores resultados en los experimentos anteriores y los resultados fueron ofrecidos de la misma forma que en el experimento anterior.

- **Simulación final con estadísticas enfrentadas sin cuotas:** Este experimento fue muy similar a los dos anteriores y únicamente varió respecto a ellos en que en los ficheros de datos (entrenamiento y test), cada ejemplo tenía como atributos estadísticas de los jugadores que disputaban un partido enfrentadas, es decir, en todas las estadísticas numéricas se hizo la resta de los datos del jugador 1 menos los del jugador 2. El algoritmo de clasificación aplicado también fue el que de media tuvo mejores resultados en los experimentos anteriores y los resultados fueron ofrecidos de la misma forma que en el experimento anterior.

- **Simulación final con estadísticas enfrentadas con cuotas:** Este experimento es muy similar al anterior y únicamente varió respecto a él en que los ficheros de datos (entrenamiento y test), además de tener como atributos el resultado de enfrentar las estadísticas de los jugadores que disputaban un partido y algunos datos del partido, fueron añadidas las cuotas a favor de cada uno de los jugadores que disputaban cada partido. El algoritmo de clasificación aplicado también fue el que de media tuvo mejores resultados en los experimentos anteriores y los resultados fueron ofrecidos de la misma forma que en el experimento anterior.

Cabe destacar que en todos estos experimentos no se probó la variación de los distintos parámetros con los que cuentan los algoritmos de clasificación, por tanto, se

usaron en casi todos los casos los parámetros por defecto establecidos por la herramienta utilizada para el desarrollo de los experimentos (Weka). Esta decisión fue tomada así debido a la excesiva carga computacional que podría ocasionar el probar distintas combinaciones de parámetros ya que, como se ha comentado anteriormente hay experimentos que tardaron varios días en terminar su ejecución.

6.3 Construcción de los modelos

En este apartado se llevará a cabo la descripción de los experimentos listados en el apartado anterior, para la realización de los mismos se utilizó la herramienta Weka. Esta herramienta da la posibilidad de configurar gran cantidad de parámetros, en general en este proyecto se trabajó con los parámetros por defecto debido a que ajustar de forma óptima todos los parámetros podría haber ocasionado un retraso elevado en el desarrollo del proyecto, pues la ejecución de los experimentos del proyecto se alargaba en ocasiones durante varios días.

A continuación, se describen detalladamente los distintos experimentos llevados a cabo en el proyecto, se describirán los parámetros configurables más importantes y se detallarán los modelos obtenidos informando de su interpretación y documentando las dificultades encontradas.

Los experimentos se dividieron en grupos dentro de los cuales había varios experimentos relacionados entre sí.

6.3.1 Experimento 1: Sólo datos estadísticos

Este experimento fue el primero llevado a cabo en el proyecto, sirvió de toma de contacto con la herramienta Weka, que fue lo más simple, y con la programación de consultas a la base de datos para formar los ficheros de datos con el formato adecuado para Weka. La programación de consultas se llevó a cabo a través de pequeños programas en Java, las primeras consultas programadas requirieron un tiempo elevado de trabajo pero en los experimentos finales, debido a la experiencia adquirida, fue una tarea mucho menos compleja.

6.3.1.1 Conjunto de datos

El fichero de datos a través del cual se llevó a cabo este experimento se formó mediante consultas a la base de datos. Cada ejemplo del fichero correspondía con cada uno de los partidos almacenados en la tabla `Partidos`, estos partidos pertenecían a un torneo y eran jugados en todos los casos por dos jugadores.

Partiendo de esta información base se obtuvo información de los dos jugadores y del torneo al que pertenecía el partido. De esta forma se iba completando cada ejemplo a través de distintas consultas que iban dando valor a cada uno de los atributos. En total se contaba con 57.400 ejemplos con 154 atributos cada uno, seis de tipo nominal, uno de tipo cadena de caracteres, que será eliminado por no aportar información relevante y para poder ejecutar algunos algoritmos de clasificación que no permiten este tipo de atributos, y el resto eran de tipo numérico. Estos atributos eran iguales a los utilizados en muchos experimentos por lo que para no repetir la descripción en varios apartados, son detallados en la Tabla 87. Muchos atributos se detallan en parejas puesto que su significado es el mismo, no así su valor, ya que son datos del jugador 1 y del jugador 2, por ejemplo, `edadJ1` es similar en significado a `edadJ2`, cada uno representa la edad de uno de los dos jugadores que disputaban los partidos, por tanto sólo habrá una descripción para los dos atributos. Para ver en detalle todos los atributos del fichero de datos ver el *Anexo B: Atributos del Experimento 1*.

6.3.1.2 Algoritmos de clasificación

Una vez generado el fichero de datos con el formato requerido por Weka se estaba en disposición de aplicar los distintos algoritmos de clasificación. Como ya se mencionó, en este experimento, se realizó validación cruzada de diez subconjuntos. Los algoritmos de clasificación aplicados a este conjunto de datos son los siguientes:

ZeroR

Este algoritmo no tiene ningún parámetro configurable debido a que lo único que hace es dar el valor de la clase más frecuente a todos los ejemplos. El resultado obtenido con este algoritmo fue del 50,11% de acierto.

OneR

Sólo dispone de un parámetro configurable, "`minBucketSize = 6`", que es el número mínimo de valores que debe tener un atributo. Esta opción es relevante cuando se trabaja con datos numéricos, como en este caso en el que la mayoría de atributos eran de este tipo. El atributo elegido para definir las reglas en las que basarse para seleccionar una u otra clase fue el atributo `CabezaSerieJ1`. El modelo construido por el algoritmo de clasificación fue el siguiente.

```
Si CabezaSerieJ1 = {0, 18, 27, 33, 4q, 5LL, 6SE, 8WC, ALT, LL, PR, SE, WC}
Entonces ganador = 2
Sino ganador =1
```

El resultado obtenido con este algoritmo de clasificación fue del 59,32% de acierto.

DecisionTable

Para el entrenamiento de la técnica de tabla de decisión (*Decison Table*), se utilizó como método de búsqueda, para encontrar buenas combinaciones de atributos para la tabla de decisión, el algoritmo `best-first` (el primero el mejor) con los

parámetros por defecto. Estos parámetros son los siguientes: "`direction = Forward`" que indica la dirección de búsqueda, en este caso hacia adelante, "`lookupCacheSize = 1`" para indicar el tamaño máximo de la cache de búsqueda de subgrupos, "`searchTermination = 5`" es el valor para el *backtracking* o marcha atrás y por último `starSet` que se deja vacío e indicaría el punto de partida para la búsqueda.

En cuanto al resto de parámetros de la tabla de decisión el parámetro "`crossVal = 1`" establece el número de subconjuntos para la validación cruzada en 1. Por otra parte se ofrece la opción de mostrar las reglas generadas con el parámetro "`displayRules = false`". El parámetro "`evaluationMeasure = Default`" en este caso permite variar la medida utilizada para evaluar el rendimiento de las combinaciones de atributos utilizados en la tabla de decisión. El último parámetro "`useIBK = false`" indica que si un dato no encuentra correspondencia con ninguna regla, se asigna a la clase mayoritaria; en caso de haber sido puesto a true, el dato se asignaría a la clase cuyas reglas estén más próximas.

El resultado obtenido con este algoritmo mejoró como era de esperar los resultados de `ZeroR` y `OneR` alcanzando el 64,02% de acierto

NaiveBayes

Pocos parámetros configurables dispone el algoritmo `NaiveBayes`. Los tres parámetros disponibles sólo cuentan con la opción activar/desactivar. El valor por defecto de estos parámetros es desactivado y así es como se llevó a cabo el experimento.

El primero de los tres parámetros es `displayModelInOldFormat` que permite mostrar el modelo usando el viejo formato de salida, este formato es mejor cuando la clase tiene muchos valores (no es el caso pues sólo había dos valores para la clase), el formato nuevo (por defecto) es mejor cuando hay pocas clases y muchos atributos. El segundo parámetro es `useKernelEstimator` que usa el estimador *Kernel* para los atributos numéricos en vez de usar una distribución normal. Por último, el parámetro `useSupervisedDiscretization` que permite usar discretización supervisada para convertir atributos numéricos en nominales.

El resultado obtenido con este algoritmo fue del 61,23% de acierto.

C4.5

Como en el resto de experimentos se optó por poner los parámetros por defecto de la herramienta Weka. El algoritmo `C4.5` es el que más parámetros configurables posee hasta el momento. El primero de ellos es "`binarySplits = false`" en caso de haberlo activado las divisiones sobre las variables discretas/nominales serían siempre binarias en la construcción de los árboles. El siguiente parámetro es "`confidenceFactor = 0.25`" y es el factor de confianza utilizado para la poda, cuanto más pequeño sea más podas serán realizadas. El parámetro "`minNumObj = 2`" es el número mínimo de instancias por hoja. A continuación, se establece

"numFolds = 3" el cual define el número de subconjuntos en que hay que dividir el conjunto de ejemplos para, el último de ellos, emplearlo como conjunto de test si se activase el parámetro "reducedErrorPruning = false" que como se ve no fue activado, en caso de haber sido activada esta opción, el proceso de poda no sería el propio del C4.5, sino que el conjunto de ejemplos se dividiría en un subconjunto de entrenamiento y otro de test, de los cuales este último serviría para estimar el error de poda. El siguiente parámetro "saveInstanceData = false" al no ser activado una vez finalizada la creación del árbol de decisión se eliminan todas las instancias que se clasifican en cada nodo, que hasta el momento se mantenían almacenadas. Con el parámetro "seed = 1" se indica la semilla usada para la generación de números aleatorios, sólo tiene efecto si reducedErrorPruning hubiese estado activado. El parámetro "subtreeRaising = true" permite realizar la poda con el proceso *subtree raising.* Para terminar los parámetros "unpruned = false" que al no estar activado realiza la poda del árbol y por último "useLaplace = false" que de estar activado al intentar predecir la probabilidad de que una instancia perteneciera a una clase, se emplearía el suavizado de Laplace.

El modelo generado por este algoritmo fue un árbol de decisión de grandes dimensiones, en modo texto ocupa más de 10.000 líneas y en concreto cuenta con 8.233 hojas siendo el tamaño del árbol 10.560. Este modelo se encuentra almacenado en los ficheros de resultados del Experimento1 adjuntos en el CD del proyecto. A continuación, se muestra un fragmento del árbol de decisión generado en el que se puede observar su raíz.

```
C4.5 pruned tree
------------------

progresionPuntosJ2 <= 1155
|   posJ1 <= 209
|   |   progresionPuntosJ1 <= 1645
|   |   |   posJ2 <= 209
|   |   |   |   ganadosJ2 <= 261
|   |   |   |   |   ganadosAnioAnteriorJ1 <= 70
|   |   |   |   |   |   posJ1 <= 37
|   |   |   |   |   |   |   posJ2 <= 66
|   |   |   |   |   |   |   |   posJ2 <= 21
|   |   |   |   |   |   |   |   |   ganadosSuperficieJ1 <= 110
|   |   |   |   |   |   |   |   |   |   perdidosAnioAnteriorJ2 <= 20
|   |   |   |   |   |   |   |   |   |   |   posPasadoJ2 <= 1: 2 (23.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   posPasadoJ2 > 1
|   |   |   |   |   |   |   |   |   |   |   |   jugador1 <= 11
```

El resultado obtenido con el árbol de decisión C4.5 fue del 63,24% de acierto

IB1

Este algoritmo de clasificación no tiene parámetros configurables y el resultado obtenido con él no fue nada satisfactorio ya que únicamente obtuvo un 52,71% de acierto y además tuvo el problema de ser el algoritmo más lento de todos los probados en este experimento.

AdaBoostM1

Como ya se explicó en el apartado *6.1.1 Clasificación* este algoritmo pertenece al conjunto de meta clasificadores. En este algoritmo el primer parámetro que se configura es el clasificador base con el cual se desea crear los modelos base, en este caso se eligió el clasificador `C4.5` con los parámetros por defecto explicados anteriormente en este mismo experimento, así pues el parámetro fue `"classifier = C4.5 -C 0.25 -M 2"`. El siguiente parámetro a configurar fue `"numIterations = 10"` que indica el número de iteraciones máximas, es decir, de modelos base. También se encuentra en este algoritmo el parámetro `"seed = 1"` que indica la semilla usada para la generación de números aleatorios. A continuación, está el parámetro `"useResampling = false"` que en caso de estar activado usaría re muestreo en lugar de reponderación. Por último, el parámetro `"weightThreshold = 100"` que indica el umbral de peso para la poda de pesos.

Como resultado se obtienen 10 árboles de decisión similares al árbol de decisión generado con el algoritmo de clasificación `C4.5`, estos árboles pueden ser consultados en los ficheros de resultados del Experimento 1 adjuntos en el CD del proyecto. A continuación, en la Tabla 13 se muestran los datos globales de cada uno de los árboles:

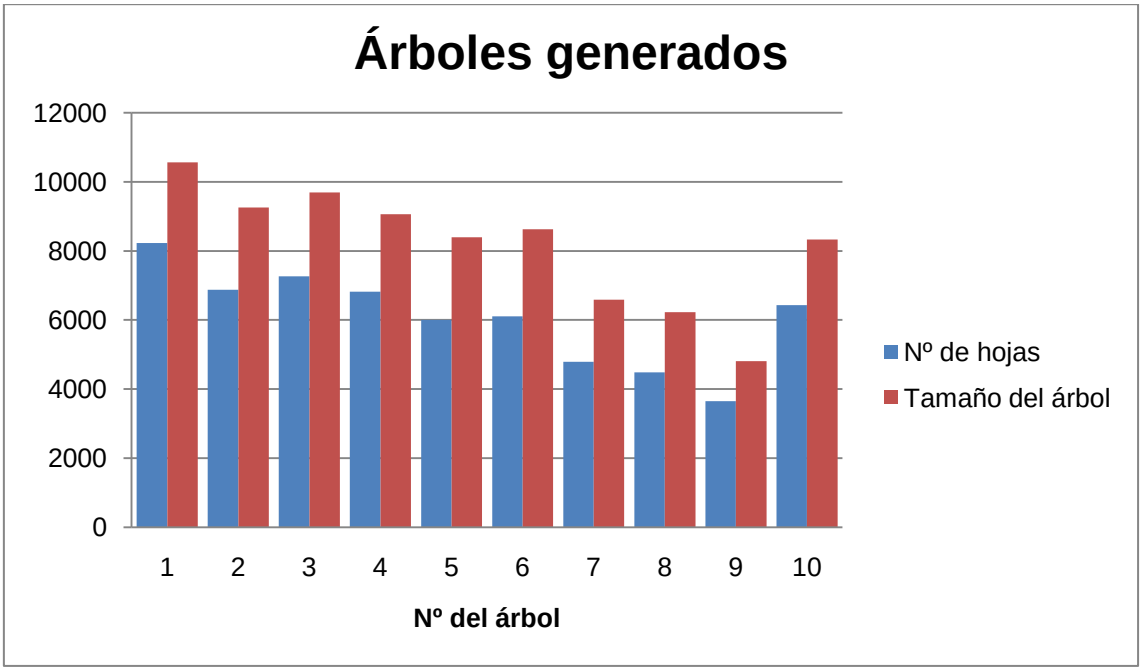


Tabla 13: Árboles generados con el algoritmo `AdaBoostM1` en el experimento 1.

El resultado obtenido con este algoritmo fue del 62,27 % de acierto.

Bagging

Como el algoritmo anterior, pertenece al conjunto de meta clasificadores. Los parámetros de este algoritmo son similares a los del algoritmo anterior (`AdaBosstingM1`), de esta forma se encuentran los parámetros `classifier`, `numIterations y seed` que tomarán exactamente los mismos valores que en el

punto anterior por tanto se tiene que "`classifier = C4.5 -C 0.25 -M 2`", "`seed = 1`" y "`numIterations = 10`". Además, el algoritmo cuenta con otros dos parámetros, el primero "`bagSizePercent = 100`" indica el porcentaje de casos seleccionados para generar las muestras *boostrap*. El último parámetro "`calcOutOfBag = false`" indica si se quiere calcular el error fuera de la bolsa.

El modelo resultante obtenido fueron nuevamente diez árboles de decisión, estos árboles pueden ser consultados en los ficheros de resultados del Experimento 1 adjuntos en el CD del proyecto. A continuación, en la Tabla 14 se muestran los datos globales de cada uno de los árboles:

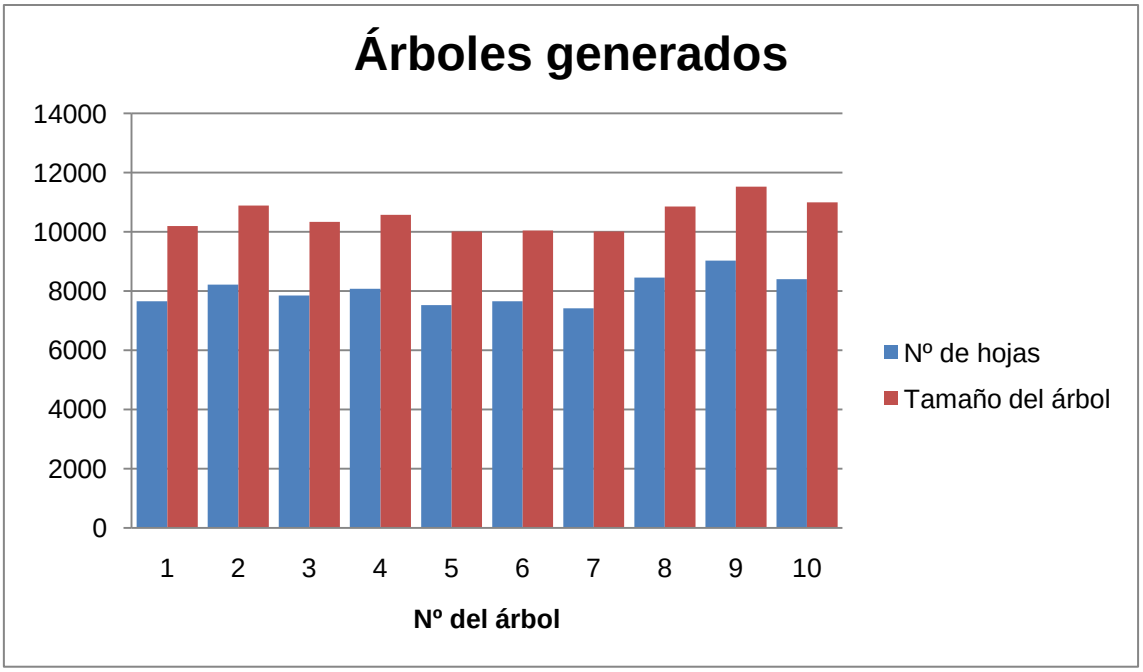


Tabla 14: Árboles generados con el algoritmo `Bagging` en el experimento 1.

El resultado obtenido con este algoritmo fue del 64,88% de acierto.

6.3.1.3 Resultados del experimento

En este apartado se hace una evaluación global del experimento comparando los resultados ofrecidos por los distintos algoritmos de clasificación aplicados en el Experimento 1. En la Tabla 15 se muestra un gráfico con el resumen de los resultados.

Los resultados obtenidos en este experimento sirvieron como toma de contacto con el dominio del proyecto. Estos resultados hacen ver que el problema no es ni mucho menos trivial ya que el mejor resultado obtenido en esta primera toma de contacto no supera el 65% de acierto. Un partido de tenis sólo puede tener un ganador y no se puede empatar, partiendo de esta base se podría suponer que eligiendo un ganador aleatoriamente se tendría el 50% de posibilidades de acertar, esto es lo que hace el algoritmo `ZeroR` que obtiene el 50,11% de acierto. Este resultado sirvió de punto de partida para el resto de algoritmos y debería ser superado en todos los

casos, de lo contrario indicaría que algo se estaría haciendo mal o que el algoritmo no era el adecuado.

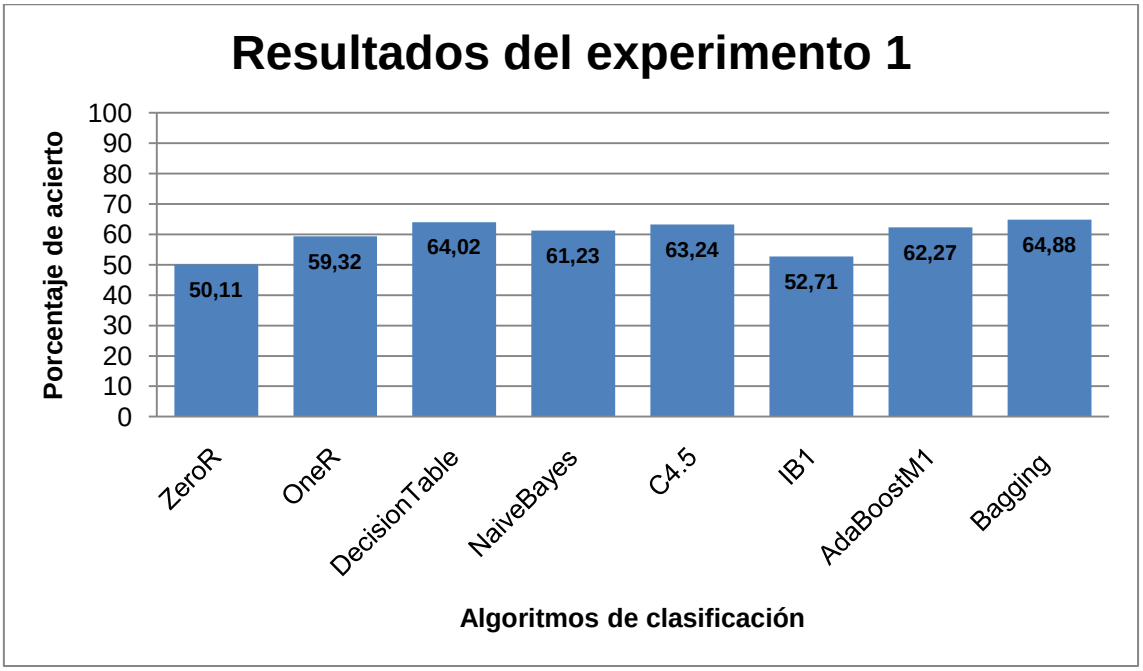


Tabla 15: Resumen de resultados del experimento 1.

Llama la atención el buen resultado obtenido con el algoritmo `DecisionTable` que no por ser uno de los más simples obtuvo peores resultados. Este algoritmo únicamente tuvo en cuenta uno de los atributos para clasificar cada uno de los ejemplos y aún así es el segundo algoritmo que mejor resultado obtuvo sólo superado por `Bagging`. Esto puede ser un indicio de que pueden existir atributos redundantes en el fichero de datos, en otros experimentos se aplicaron algoritmos de selección de atributos para comprobar si esto ocurría.

Sorprende el resultado del algoritmo `IB1` por ser malo y por ser uno de los algoritmos que más tiempo requirió para en su ejecución. Es posible que este mal resultado fuese debido a que pocos ejemplos se parecían entre sí puesto que, los jugadores no eran iguales unos a otros, todos tenían sus diferencias y por lo tanto los ejemplos que posiblemente se pareciesen entre sí son los que representaban partidos entre dos jugadores (lo mismos), pero en distintos enfrentamientos (torneos o fechas). El problema es que era tal la cantidad de jugadores que había que normalmente pocas veces un jugador repetía rival a lo largo de un año.

Por último, señalar que el mejor resultado fue conseguido por el algoritmo de clasificación `Bagging`, eso sí, este algoritmo genera como resultado un modelo basado en árboles de decisión bastante complejo pues cuentan con un gran número de hojas y nodos intermedios.

También resaltar que uno de los problemas de este experimento fue que el fichero de datos contaba con demasiados ejemplos y cada ejemplo estaba compuesto por más de 150 atributos, lo cual dio como resultado un fichero de datos muy grande.

Esto provocó que la ejecución de cada una de las pruebas durase varios días hasta obtenerr el resultado final.

6.3.2 Experimento 2: Sólo apuestas en vivo

El primer experimento marcó el camino de lo que serían el resto de experimentos. Como ya se señaló en puntos anteriores los datos fueron obtenidos de dos fuentes distintas. El fichero de datos del primer experimento sólo contenía datos obtenidos de la base de datos de *OnCourt*, en este segundo experimento se trabajó con los datos ofrecidos por la otra fuente de datos utilizada para formar la base de datos final del proyecto, es decir, los datos obtenidos de *Betfair*.

Cada una de las líneas de los datos ofrecidos por *Betfair*, se podía clasificar según si era o no a una apuesta realizada en vivo observando el campo `En_juego` de la tabla `Betfair` de la base de datos. A través de este experimento se trató de predecir el resultado de las apuestas en vivo teniendo en cuenta el número de apuestas, el volumen de las mismas y la cuota a la que se apostaba. Se trató de ver, por ejemplo, si se puede predecir que una apuesta va a ser ganadora según la cuota que tenga o el volumen que se haya jugado a esa cuota.

6.3.2.1 Conjunto de datos

El fichero de datos creado para este experimento se generó a través de consultas a la tabla `Betfair` de la base de datos. Cada ejemplo del fichero correspondía con cada una de las apuestas pertenecientes a un partido y realizadas en vivo de la tabla `Betfair`. Para realizar el filtrado de apuestas en vivo se recorrió la tabla `Betfair` buscando las líneas en las que el campo `Partido` no fuera `null`, es decir, que la apuesta perteneciera a un partido, y además, el campo `En_juego` fuera igual a `IP` (del inglés *In Play*).

En total se formó un fichero con 2.340.325 ejemplos cada uno de los cuales contaba con cuatro atributos, dos de tipo real, uno de tipo numérico y otro nominal. Todos estos atributos fueron obtenidos de la tabla de datos `Betfair` que a pesar de contar con más atributos, estos fueron considerados irrelevantes para este experimento. Para ver los atributos que formaban cada uno de los ejemplos del experimento ver *Anexo C: Atributos del Experimento 2*.

6.3.2.2 Algoritmos de clasificación

Una vez generado el fichero de Weka se procedió a la realización del experimento aplicando los mismos algoritmos que en el experimento 1 y con los mismos parámetros. En principió el experimento fue diseñado para realizarse con validación cruzada, sin embargo, surgió un problema, ya que, al disponer de tantas

instancias la aplicación Weka producía un error interno que impedía llevar a cabo la ejecución del experimento. Así pues, se tomó la determinación de que en vez de realizar validación cruzada se dividirían los ejemplos aleatoriamente en dos ficheros, uno de entrenamiento que estaba formado por 1.545.755 ejemplos y otro de test formado por 794.564 ejemplos, es decir, el de entrenamiento contaba con el 66,66% de los ejemplos y el de test con el 33,33% restante. Para dividir los ejemplos en los ficheros de entrenamiento y test se realizó un programa en Java que separó los ejemplos aleatoriamente utilizando el método `random` de la clase `Math`. De esta forma se recorrieron todos los ejemplos enviando cada uno de ellos al fichero de entrenamiento en el 66% de los casos y al de test en el 33% restante.

Los algoritmos de clasificación aplicados en este experimento fueron exactamente los mismos y con los mismos parámetros que los aplicados en el Experimento 1, es por ello que no se volverá a repetir la descripción de los parámetros. A continuación, se listan los algoritmos aplicados con los comentarios más relevantes en cada caso:

ZeroR

En este caso la clase mayoritaria de los ejemplos era la clase `T`, es decir, en el fichero de entrenamiento había más apuestas/ejemplos que resultaban ganadores, de los que resultaban perdedores. Por tanto, el algoritmo en el fichero de test señalaba cada uno de los ejemplos como ganador obteniendo un resultado total del 52,45% de acierto y siendo la matriz de confusión la siguiente:

```
=== Confusion Matrix ===

      a      b   <-- classified as
 416764      0 |      a = T
 377800      0 |      b = F
```

Esta matriz indica que en el fichero de test hay 416.764 ejemplos que son de la clase `T`, es decir, que han resultado apuestas ganadoras y en los cuales la predicción del algoritmo ha sido correcta. Y por otra parte hay 377.800 ejemplos que siendo de la clase `F` han sido clasificados como `T`, es decir, su predicción fue incorrecta. Por tanto, como era de esperar con este algoritmo se obtuvo un resultado similar al de tirar una moneda al aire y adivinar si es cara o cruz.

OneR

Ya se señaló en apartados anteriores que el algoritmo `OneR` elige un atributo a partir del cual genera una serie de reglas que clasifican cada uno de los ejemplos. En este caso sólo se disponía de cuatro atributos, de los cuales el atributo `Ganada` es descartado por ser la clase. Antes de ejecutar el algoritmo por sentido común se esperaba que el atributo elegido para generar las reglas fuese la cuota pues está relacionada con el resultado final del evento ya que la cuota es inversamente proporcional a la probabilidad del resultado.

Después de la ejecución del algoritmo se comprobó que, efectivamente el atributo elegido fue Cuota siendo la regla generada la siguiente:

```
Cuota:
  < 1.955 -> T
  < 1.9649999999999999  -> F
  < 1.975 -> T
  >= 1.975       -> F
```

El resultado obtenido al aplicar esta regla al fichero de test fue del 73,81% de acierto.

DecisionTable

El resultado obtenido con este algoritmo de clasificación fue similar al del algoritmo `OneR` con un 73,81% de acierto. Sin embargo, se podría considerar que el resultado es peor puesto que a pesar de tener el mismo porcentaje de acierto las reglas generadas, para obtener estos aciertos, fueron mucho más complicadas que las obtenidas con el algoritmo `OneR`. En total se generan 1.105 reglas que contrastan con las cuatro generadas por el algoritmo anterior. Además, se tuvieron en cuenta dos atributos (`Cuota y Numero_Apuetas`) en lugar de uno.

NaiveBayes

Este algoritmo no consiguió buenos resultados en este experimento y únicamente obtuvo un porcentaje de acierto del 59,57%.

C4.5

El algoritmo `C4.5` obtuvo un resultado todavía más simple que el obtenido por el algoritmo `OneR`. En este caso el modelo de clasificación fue un árbol de decisión con únicamente 2 hojas y un tamaño del árbol de 3. A continuación, se muestra el árbol decisión tal y como se obtuvo en Weka:

```
C4.5 pruned tree
------------------

Cuota <= 1.99: T (875689.0/234641.0)
Cuota > 1.99: F (670066.0/169807.0)

Number of Leaves  :     2

Size of the tree :      3
```

Aplicando este árbol de decisión al fichero de test se consiguió un porcentaje de acierto del 73,83% siendo el mejor obtenido hasta ahora.

IB1

No fue posible la aplicación del algoritmo pues se produjeron errores en tiempo de ejecución que impidieron la ejecución del algoritmo y la obtención de los resultados.

AdaBoostM1

No fue posible la aplicación del algoritmo por el mismo motivo que en el algoritmo IB1, se produjeron errores en tiempo de ejecución que impidieron la ejecución del algoritmo y la obtención de los resultados.

Bagging

El resultado obtenido fue similar al obtenido anteriormente con el algoritmo `C4.5`, sin embargo, el modelo de clasificación fue mucho más complejo pues contaba con diez árboles de decisión, siendo alguno de ellos idéntico al generado por el algoritmo `C4.5`. El tamaño de los árboles puede observarse a través del gráfico de la Tabla 16.

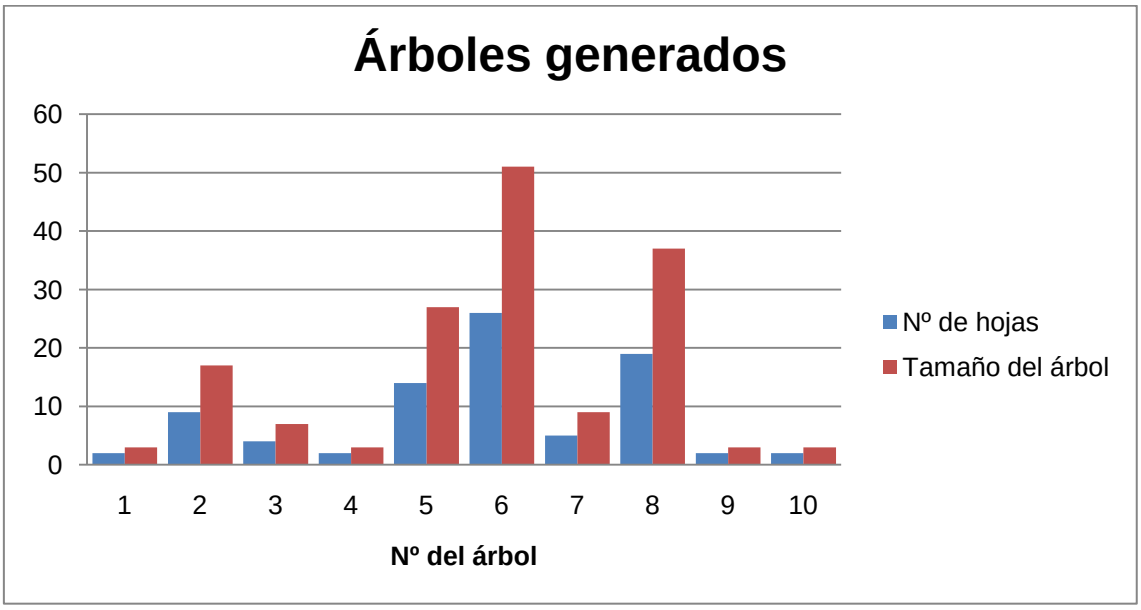


Tabla 16: Árboles generados con el algoritmo `Bagging` en el experimento 2.

El resultado total obtenido fue del 73,83% de acierto.

6.3.2.3 Resultados del experimento

A continuación, en la Tabla 17 se muestran todos los resultados obtenidos a lo largo del experimento.

Los resultados obtenidos en este experimento fueron alentadores ya que, si únicamente con cuatro atributos se conseguían estos porcentajes de acierto, si a cada uno de los ejemplos se le añadiera información de los jugadores que lo disputan se podría suponer que los resultados podrían mejorar. Cabe destacar que, como en el anterior experimento, los mejores resultados vinieron dados por los algoritmos `DecisionTable, C4.5 y Bagging`. Sin embargo, sorprendía la inclusión en este caso del algoritmo `OneR` que funcionó bien en este caso, esto pudo ser debido al escaso número de atributos y a que uno de ellos, la cuota, tenía un peso mucho mayor que los otros atributos en cuanto a su importancia en el resultado final.

En comparación con los resultados del experimento anterior también hay que destacar el mal resultado del algoritmo `NaiveBayes` que en este caso estaba muy lejos de los mejores resultados.

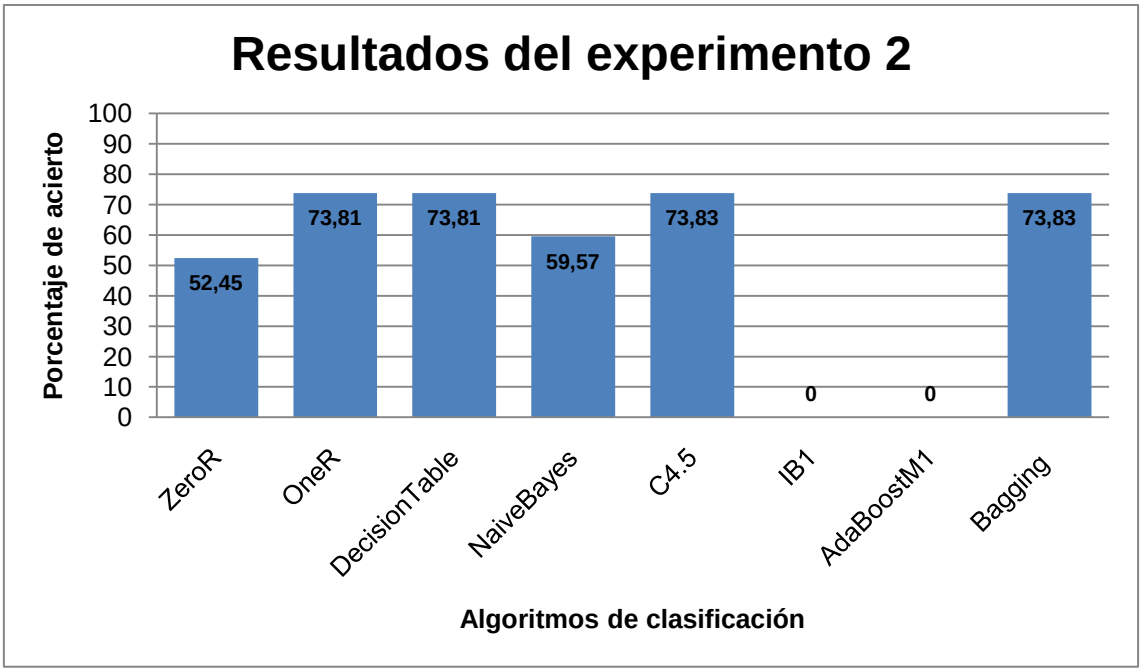


Tabla 17: Resumen de resultados del experimento 2.

El siguiente experimento fue similar a este con la salvedad de que realizará la predicción con datos de apuestas de antes del comienzo de los partidos, por tanto, los resultados obtenidos en principio debían ser peores. Esto es debido a que en una apuesta en vivo se tiene información acerca de cómo va el marcador y quién está más cerca de la victoria, es decir, se dispone de información muy útil para predecir el resultado final. Esta información viene reflejada en la cuota a favor de cada uno de los jugadores, cuanto más pequeña sea la cuota más probable será que resulte ganadora la apuesta. Por ejemplo, si en un partido de Rafael Nadal contra Roger Federer antes del comienzo del partido se pagan tanto la victoria de Nadal como la de Federer a 2 € por euro apostado y al cabo de una hora de partido la victoria de Nadal se paga a 1,5€ y la de Federer a 3€ es muy probable que sea debido a que en ese instante Nadal vaya por delante en el marcador.

6.3.3 Experimento 3: Sólo apuestas pre inicio

Este experimento fue muy similar al Experimento 2, sólo se diferenció en que los ejemplos fueron todos de apuestas realizadas antes del comienzo de los partidos. Además, se añadió un atributo llamado `EnJuego` que podría tener los valores `"P"` o `"N"`. Este atributo indica que la apuesta fue realizada antes del comienzo del partido porque en vivo no era posible `(N)`, ya que ese partido no era ofrecido por *Betfair* en la modalidad de apuestas en vivo, o porque fue decisión del apostante el no apostar en vivo `(PE)` si ofreciéndose el partido en la modalidad de apuestas en vivo.

Además, se volvió a utilizar validación cruzada de diez subconjuntos pues en este caso se disponía de menos de la cuarta parte de los ejemplos disponibles para el Experimento 2.

6.3.3.1 Conjunto de datos

Como en el experimento anterior el fichero de datos creado para este experimento se generó a través de consultas a la tabla `Betfair` de la base de datos. Cada uno de los ejemplos del fichero correspondía con una apuesta perteneciente a un partido y realizada antes del comienzo del mismo, ya sea porque no se podía apostar en vivo a ese partido o porque pudiéndose apostar en vivo se realizó antes del comienzo del partido. Para realizar el filtrado de apuestas se recorrió la tabla `Betfair` buscando las filas en las que el campo `Partido` no estuviera vacío y además el campo `En_juego` fuera igual a `NI` (del inglés *Not In-play*) o `PE` (del inglés *Pre-Event*).

En total se generó un fichero con 806.908 ejemplos cada uno de los cuales contaba con cinco atributos, dos de tipo real, dos de tipo nominal y otro de tipo numérico. Estos atributos fueron todos obtenidos de la tabla `Betfair`, que a pesar de contar con más atributos, estos no fueron tenidos en cuenta para este experimento. Para ver los atributos que forman cada uno de los ejemplos del experimento ver *Anexo D: Atributos del Experimento 3*.

6.3.3.2 Algoritmos de clasificación

Este experimento fue muy similar al anterior y por tanto se ejecutaron los mismos algoritmos de clasificación.

ZeroR

Con la aplicación de este algoritmo se comprobó que la clase mayoritaria de este experimento era la clase `F`, es decir, en el fichero de datos había más ejemplos que pertenecían a apuestas perdedoras de los que pertenecían a apuestas ganadoras. Por tanto, el algoritmo en cada uno de los diez subconjuntos de test señalaba cada uno de los ejemplos como perdedor, acertando de media en un 56,23% de los casos. La matriz de confusión obtenida es la siguiente:

```
=== Confusion Matrix ===

      a      b   <-- classified as
      0 353111 |      a = T
      0 453797 |      b = F
```

A través de la matriz se puede saber que en total había 453.797 ejemplos que pertenecían a la clase `F`, es decir, que resultaban apuestas perdedoras y los cuales fueron clasificados de forma correcta por el algoritmo. Este resultado contrastaba en cierta medida con el obtenido en el experimento anterior por el mismo algoritmo pues es casi un 4% mayor, esto pudo ser debido a que en las apuestas realizadas en vivo

se conocía información del resultado en cada momento del partido, lo que al final conllevaba a que el apostante acertase el resultado en la mayoría de los casos, no como ocurría en las apuestas pre partido en las que el apostante perdía la apuesta más veces de las que la ganaba.

OneR

Como en el experimento anterior y, como era de esperar, el atributo elegido por el algoritmo para establecer una regla y clasificar con ella cada uno de los ejemplos fue el atributo `Cuota` que como se señaló antes es inversamente proporcional a la probabilidad del resultado.

El modelo generado por el algoritmo fue muy simple y consistía en una única regla que es mostrada a continuación:

```
Cuota:
  < 1.9649999999999999  -> T
  >= 1.9649999999999999 -> F
```

De esta forma si la cuota a la que se paga un determinado evento es menor a 1,9649999 habría que apostar a favor de ese evento pues el modelo de clasificación indica que resultará ganadora. En caso contrario habría que apostar en contra pues el modelo indica que la apuesta resultará perdedora.

El resultado final obtenido con el algoritmo fue de un 71,56% de acierto. Este resultado es algo inferior al obtenido en el Experimento 2 debido a que en una apuesta pre inicio se tiene menos información del partido que en las apuestas en vivo.

DecisionTable

Con la aplicación de este algoritmo se obtuvo un resultado muy similar al obtenido con el algoritmo `OneR`. En este caso el porcentaje de acierto fue ligeramente superior con un 71,57% de acierto. Las reglas generadas para clasificar la clase también estaban basadas únicamente en el atributo `Cuota`, sin embargo, éstas eran más complejas en cuanto a número. En el algoritmo `OneR` sólo se dividía la cuota en dos para clasificar la clase mientras que este algoritmo dividió la cuota en un total de 44 rangos siendo el modelo completo el siguiente:

```
Rules:
=======================
Cuota           Ganada
=======================
'(-inf-1.025]'  T
'(25.5-47]'     F
'(47-inf)'      F
'(16.25-25.5]'  F
'(1.025-1.045]' T
'(8.9-12.25]'   F
'(12.25-16.25]' F
'(1.095-1.115]' T
'(1.055-1.095]' T
'(1.045-1.055]' T
'(4.95-5.45]'   F
'(3.425-3.625]' F
'(1.785-1.885]' T
```

```
'(1.885-1.965]' T
'(1.965-2.11]'  F
'(4.25-4.95]'   F
'(3.875-4.25]'  F
'(2.77-2.99]'   F
'(2.21-2.31]'   F
'(1.495-1.555]' T
'(1.225-1.295]' T
'(2.53-2.77]'   F
'(1.555-1.645]' T
'(1.445-1.495]' T
'(1.395-1.445]' T
'(3.225-3.425]' F
'(2.99-3.225]'  F
'(1.195-1.225]' T
'(1.165-1.195]' T
'(1.155-1.165]' T
```

```
'(1.115-1.155]' T
'(7.3-8.9]'     F
'(6.9-7.3]'     F
'(5.95-6.9]'    F
'(5.45-5.95]'   F
'(3.625-3.875]' F
'(1.345-1.395]' T
'(1.295-1.345]' T
'(1.725-1.785]' T
'(1.695-1.725]' T
'(1.645-1.695]' T
'(2.41-2.53]'   F
'(2.31-2.41]'   F
'(2.11-2.21]'   F
=======================
```

Es curioso el resultado obtenido en este caso pues si se ordenara el modelo según el valor de las cuotas de menor a mayor el resultado sería el siguiente:

```
Rules:
=======================
Cuota            Ganada
=======================
'(-inf-1.025]'  T
'(1.025-1.045]' T
'(1.045-1.055]' T
'(1.055-1.095]' T
'(1.095-1.115]' T
'(1.115-1.155]' T
'(1.155-1.165]' T
'(1.165-1.195]' T
'(1.195-1.225]' T
'(1.225-1.295]' T
'(1.295-1.345]' T
'(1.345-1.395]' T
'(1.395-1.445]' T
```

```
'(1.445-1.495]' T
'(1.495-1.555]' T
'(1.555-1.645]' T
'(1.645-1.695]' T
'(1.695-1.725]' T
'(1.725-1.785]' T
'(1.785-1.885]' T
'(1.885-1.965]' T
'(1.965-2.11]'  F
'(2.11-2.21]'   F
'(2.21-2.31]'   F
'(2.31-2.41]'   F
'(2.41-2.53]'   F
'(2.53-2.77]'   F
'(2.77-2.99]'   F
'(2.99-3.225]'  F
'(3.225-3.425]' F
```

```
'(3.425-3.625]' F
'(3.625-3.875]' F
'(3.875-4.25]'  F
'(4.25-4.95]'   F
'(4.95-5.45]'   F
'(5.45-5.95]'   F
'(5.95-6.9]'    F
'(6.9-7.3]'     F
'(7.3-8.9]'     F
'(8.9-12.25]'   F
'(12.25-16.25]' F
'(16.25-25.5]'  F
'(25.5-47]'     F
'(47-inf)'      F
=======================
```

Si se observa el modelo se puede ver que este modelo es equivalente al siguiente modelo:

```
Cuota:
  <= 1.965       -> T
  >  1.965       -> F
```

Una vez hechas las comparaciones pertinentes era evidente que, el modelo generado por el algoritmo `DecisionTable` era casi idéntico al generado por el `OneR`, sin embargo, constaba de muchas más reglas. Por tanto, para este experimento se podría decir que el algoritmo `OneR` tuvo la misma eficacia que el algoritmo `DecisionTable`, sin embargo, su solución final fue más eficiente.

NaiveBayes

Como era previsible, en este experimento este algoritmo tampoco consiguió buenos resultados obteniendo únicamente un porcentaje de acierto del 58,49%.

C4.5

El modelo generado por el algoritmo `C4.5` fue el siguiente:

```
C4.5 pruned tree
------------------

Cuota <= 2.3
|   Cuota <= 1.96: T (326721.0/101409.0)
|   Cuota > 1.96: F (72984.0/33810.0)
Cuota > 2.3: F (407203.0/93989.0)

Number of Leaves  :     3

Size of the tree :      5
```

Como se puede observar este modelo es prácticamente idéntico a los obtenidos con los algoritmos `DecisionTable` y `OneR`. También eligió al atributo `Cuota` para

generar, en este caso, el árbol de decisión que clasificaba cada uno de los ejemplos. Además, en este caso el modelo generado es equivalente al siguiente modelo de reglas:

```
Cuota:
  <= 1.96        -> T
  >  1.96        -> F
```

Si se observa este modelo y los generados anteriormente con los algoritmos `OneR` y `DecisionTable` se ve que el valor de la cuota que divide los ejemplos para que sean de una clase u otra es casi el mismo siendo 1,9649999 (`OneR`), 1,965 (`DecisionTable`) y 1,96 (`C4.5`). Por tanto, el resultado de este algoritmo tenía que ser parecido a los obtenidos con los algoritmos `OneR` y `DecisionTable`, ya que el modelo generado es muy similar en todos ellos. Al ejecutar el algoritmo se obtuvo un porcentaje de acierto del 71,56%.

IB1

No fue posible la aplicación del algoritmo pues se produjeron errores en tiempo de ejecución que impidieron la ejecución del algoritmo y la obtención de los resultados.

AdaBoostM1

El resultado obtenido fue similar al conseguido anteriormente con el algoritmo `C4.5`. Sin embargo, el modelo de clasificación fue mucho más complejo pues contaba con diez árboles de decisión, siendo uno de ellos idéntico al generado con el algoritmo `C4.5`. El tamaño de los árboles generados se detalla en la Tabla 18.

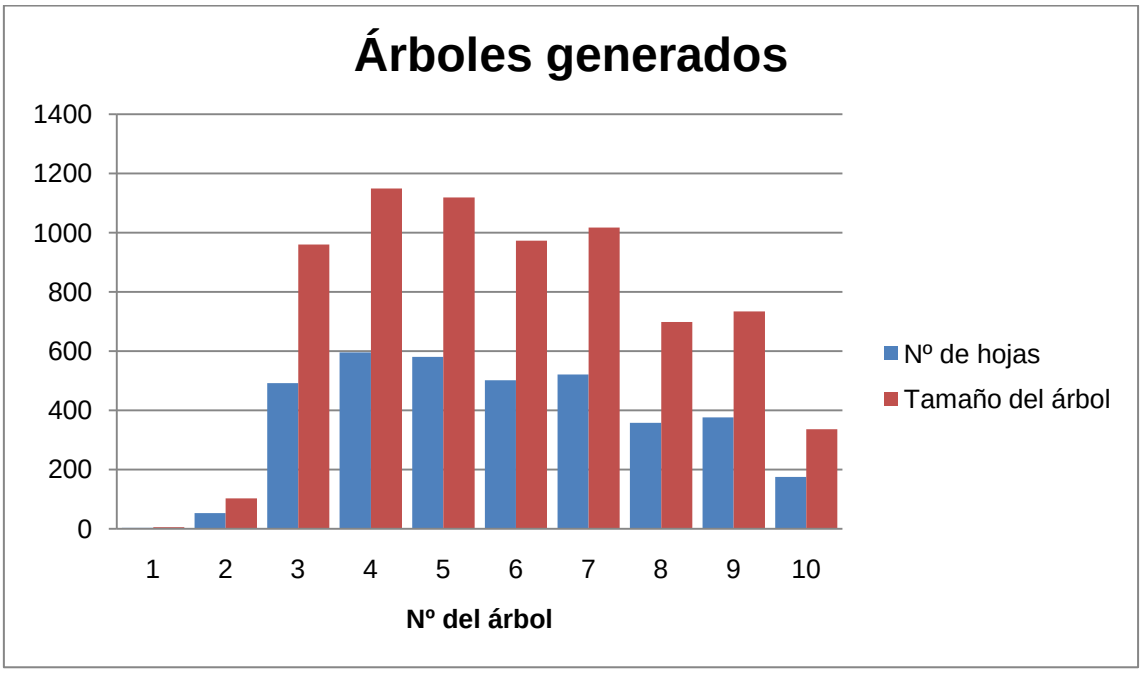


Tabla 18: Árboles generados con el algoritmo `AdaBoostM1` en el experimento 3.

Este algoritmo da un peso a cada uno de los árboles de decisión que genera, en este caso, los dos árboles que más peso tenían eran el primero con 0,92 y el segundo

con 0,61 estando el resto muy por debajo de estos valores. Esto indicaba que para este experimento los mejores modelos eran los más simples pues ambos árboles eran mucho menores que el resto de los árboles generados.

El resultado total obtenido por el algoritmo fue similar a los proporcionados anteriormente por otros algoritmos con un porcentaje de acierto del 71,46%.

Bagging

El resultado conseguido con este algoritmo fue el mejor de todos con un 71,59% de acierto. El modelo de clasificación obtenido contaba con diez árboles de decisión, en algún caso son similares a alguno de los obtenidos por el algoritmo `AdaBoostM1`. En la Tabla 19 se detalla el tamaño de cada uno de los árboles:

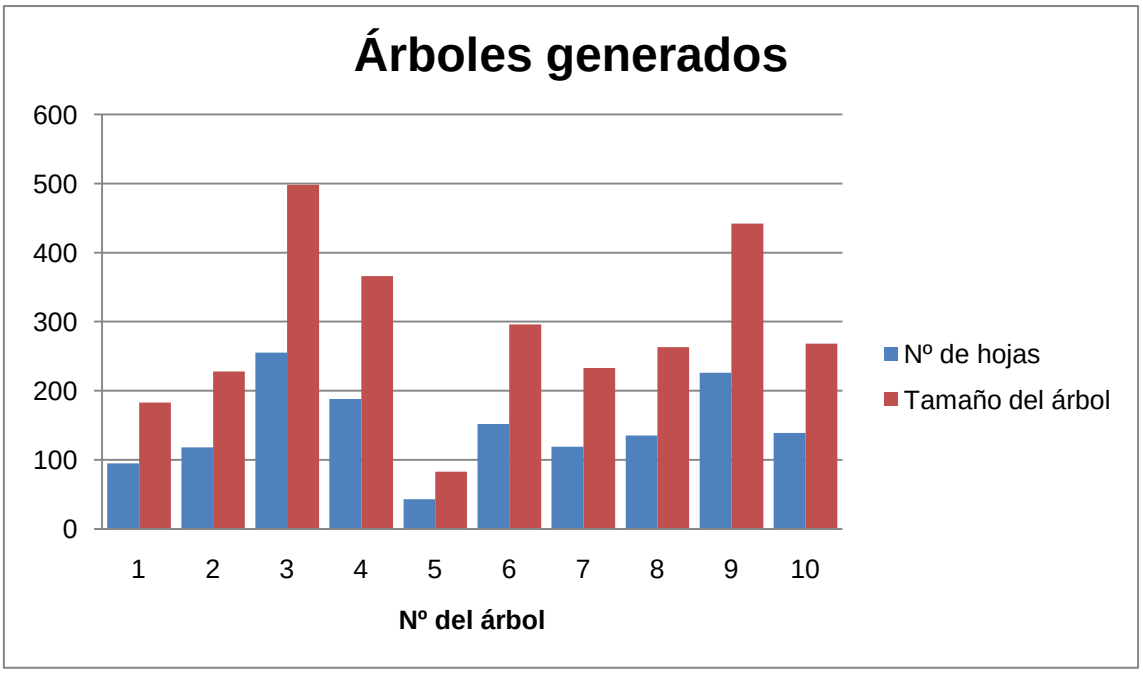


Tabla 19: Árboles generados con el algoritmo `Bagging` en el experimento 3.

6.3.3.3 Resultados del experimento

A continuación, en la Tabla 20 se muestran los resultados obtenidos a lo largo del experimento.

Como era previsible los resultados de este experimento empeoraron un poco con respecto a los del Experimento 2, esto es debido como ya se describió anteriormente a que la información dada por una cuota antes del comienzo de los partidos es menor que si el partido estuviese ya en juego. Cuando se especifica que la información es menor se está haciendo referencia a que cuando se está apostando a un resultado en vivo se conoce el marcador parcial del mismo mientras que al apostar antes del inicio del evento se desconoce ésta información.

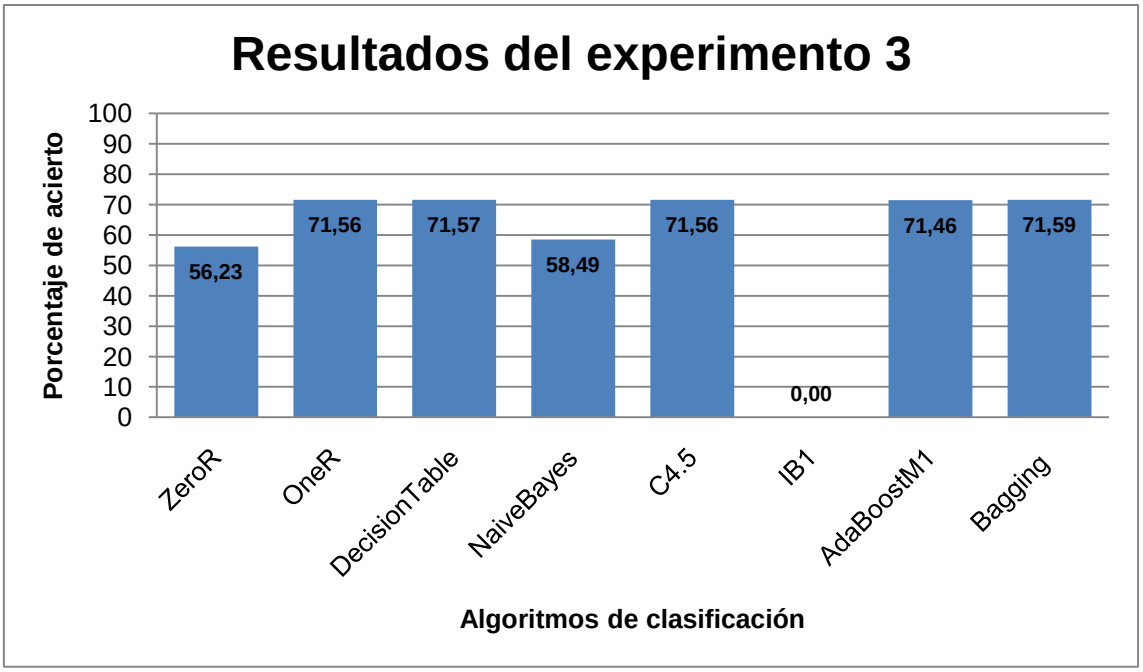


Tabla 20: Resumen de resultados del experimento 3.

Hay que destacar que en este caso todos los algoritmos funcionaron con el mismo nivel de efectividad a excepción del `ZeroR` (como es normal) y del `NaiveBayes` que empieza a dejar muestras de que los métodos únicamente estadísticos no funcionan para generar modelos de predicción de resultados de eventos deportivos.

Destacar también la sencillez de las reglas obtenidas por varios de los algoritmos y la similitud entre las mismas. De esta forma, se observa como a través de caminos tan distintos se pueden alcanzar soluciones similares en cuanto a modelos (reglas y árboles) y resultados (porcentajes de acierto).

6.3.4 Experimento 4: Experimento 1 y Selección de atributos

Este experimento fue una continuación del Experimento 1. El proceso a seguir fue la aplicación de cuatro algoritmos de selección de atributos distintos para generar cuatro subconjuntos de datos a los que se les aplicaron los mismos algoritmos de clasificación aplicados en los experimentos anteriores. De esta forma se pudo observar si había atributos innecesarios y si el seleccionar únicamente los atributos más relevantes conllevaba una mejora de los resultados.

6.3.4.1 Conjunto de datos

El conjunto inicial de datos fue exactamente el mismo que el empleado en el Experimento 1 (ver *Anexo B: Atributos del Experimento 1)*.

Para poder aplicar los algoritmos seleccionados se eliminó el único atributo de tipo cadena de caracteres del conjunto de datos (`premiosTorneo`), pues impedía la aplicación de estos algoritmos. Una vez aplicados los distintos algoritmos de selección de atributos se generaron cuatro subconjuntos de datos que serán descritos a continuación.

6.3.4.2 Algoritmos de Selección de atributos

A partir del fichero de datos con el formato de Weka generado en el Experimento 1 se generaron otros cuatro ficheros de datos mediante la aplicación de tres algoritmos distintos de selección de atributos. El objetivo era identificar aquellos atributos que tenían más peso a la hora de determinar si los ejemplos eran de una clase u otra. Los algoritmos de selección de atributos aplicados al conjunto de datos se detallan a continuación:

Subconjunto A: ChiSquaredAttributeEval + Ranker

El primer subconjunto de datos se generó seleccionando los veinte mejores atributos del ranking generado por el algoritmo `ChiSquaredAttributeEval` en combinación con el método de búsqueda `Ranker`.

El resultado obtenido con el algoritmo es una lista de los atributos con un rango, y ordenados de mayor a menor rango. De esta lista se obtuvieron los veinte de mayor rango y junto con la clase se formó el fichero de datos `A` del Experimento 4. Los veinte atributos seleccionados con su correspondiente rango asignado por el algoritmo fueron los siguientes:

```
=Attribute Selection on all input data =

Search Method: Attribute ranking.

Attribute Evaluator:
Chi-squared Ranking Filter

Ranked attributes:
2656.68425     26 CabezaSerieJ1
2381.77923     98 CabezaSerieJ2
1838.53168      4 posJ1
1791.57982      3 puntosJ1
1664.80826     76 posJ2
1586.97891     75 puntosJ2
```

```
1279.65614      7 progresionPuntosJ1
1109.57447     79 progresionPuntosJ2
1074.58398      6 posPasadoJ1
1062.1046       5 puntosPasadoJ1
1037.36482     42 ganadosJ1
1007.29733    114 ganadosJ2
 980.89743     78 posPasadoJ2
 958.41016     77 puntosPasadoJ2
 883.92086      1 jugador1
 835.66236    137 ganadosSuperficieJ2
 822.09202      8 progresionPosJ1
 784.87244     65 ganadosSuperficieJ1
 748.0199      73 jugador2
 686.9915      38 ganadosAnioJ1
```

El segundo número de la lista es el número identificativo del atributo en Weka, es asignado según el orden del atributo en el fichero de datos y en este caso era irrelevante.

Subconjunto B: InfoGainAttributeEval + Ranker

El segundo de los algoritmos de selección de atributos empleado, para la creación de otro subconjunto de datos, fue el `InfoGainAttributeEval` en combinación con el método de búsqueda `Ranker`.

El resultado de la ejecución del algoritmo fue muy similar al obtenido con el algoritmo `ChiSquaredAttributeEval` y consistió nuevamente en una lista de atributos cada uno con un rango ordenados de mayor a menor. Nuevamente el subconjunto de datos estaba formado por los veinte mejores atributos más la clase y con ellos se formó el fichero de datos `B` del Experimento 4. Los mejores veinte atributos con su rango correspondiente fueron los siguientes:

```
=Attribute Selection on all input data=

Search Method: Attribute ranking.

Attribute Evaluator:
Information Gain Ranking Filter

Ranked attributes:
 0.0341709     26 CabezaSerieJ1
 0.0305464     98 CabezaSerieJ2
 0.0238472      4 posJ1
 0.0232738      3 puntosJ1
 0.0215708     76 posJ2
 0.0205384     75 puntosJ2
```

```
 0.0164533      7 progresionPuntosJ1
 0.0142671     79 progresionPuntosJ2
 0.013856       6 posPasadoJ1
 0.013728       5 puntosPasadoJ1
 0.0132807     42 ganadosJ1
 0.0129236    114 ganadosJ2
 0.0126737     78 posPasadoJ2
 0.0124348     77 puntosPasadoJ2
 0.0114589      1 jugador1
 0.0105689    137 ganadosSuperficieJ2
 0.0104698      8 progresionPosJ1
 0.0099257     65 ganadosSuperficieJ1
 0.0098628     73 jugador2
 0.0088826     40 ganadosAnioAnteriorJ1
```

Se puede observar como la selección de atributos de los dos algoritmos presentados hasta ahora fue casi idéntica, únicamente difirieron en la selección de un atributo. En este caso el algoritmo `InfoGainAttributeEval` eligió el atributo `ganadosAnioAnteriorJ1` mientras que el algoritmo `ChiSquaredAttributeEval` eligió el atributo `ganadosAnioJ1`.

CfsSubsetEval

Con este algoritmo se probaron dos nuevos métodos de búsqueda que dieron lugar a dos subconjuntos de datos distintos. Los métodos de búsqueda elegidos fueron los siguientes:

Subconjunto C: CfsSubsetEval + RankSearch

Con este método de búsqueda se obtuvo una lista ordenada de los mejores atributos. De estos atributos los veinte mejores formaron el subconjunto de datos que generó el fichero de datos `C` del Experimento 4.

En este caso la lista de los veinte mejores atributos fue la siguiente:

```
=Attribute Selection on all input data =

Search Method:
RankSearch :

       Attribute ranking :
         26 CabezaSerieJ1
         98 CabezaSerieJ2
          4 posJ1
          3 puntosJ1
```

```
         75 puntosJ2
         76 posJ2
          6 posPasadoJ1
          5 puntosPasadoJ1
         77 puntosPasadoJ2
          7 progresionPuntosJ1
        114 ganadosJ2
         78 posPasadoJ2
         79 progresionPuntosJ2
         43 perdidosJ1
```

```
          42 ganadosJ1
         112 ganadosAnioAnteriorJ2
           8 progresionPosJ1
```

```
           1 jugador1
         115 perdidosJ2
          73 jugador2
```

Este subconjunto se diferenciaba de los dos anteriores en cuatro de los veinte atributos seleccionados.

Subconjunto D: CfsSubsetEval + GeneticSearch

El último subconjunto de datos fue almacenado en el fichero de datos `D` del Experimento 4 y se formó con los atributos seleccionados por el método de búsqueda `GeneticSearch`. Este método de búsqueda cuenta con varios parámetros configurables propios de algoritmos genéticos. En este caso, como en todos los anteriores, se dejaron los valores por defecto del programa.

Este algoritmo selecciona los mejores atributos sin ordenarlos en un ranking. En este caso fueron elegidos 71 de los 152 atributos, que junto con la clase formaron el fichero de datos `D`. Los atributos seleccionados por el algoritmo son mostrados a continuación:

```
Selected attributes:

edadJ1
puntosJ1
posJ1
puntosPasadoJ1
posPasadoJ1
progresionPuntosJ1
PPSJ1
PPGSSJ1
RGWJ1
CabezaSerieJ1
RachaJ1
ultimos5GanadosJ1
ultimos10GanadosJ1
ultimos20PerdidosJ1
ultimos50GanadosJ1
ganadosAnioJ1
ganadosAnioAnteriorJ1
ganadosJ1
perdidosJ1
ganadosPeriodoJ1
RachaSuperficieJ1
ultimos5SuperficieGanadosJ1
ultimos10SuperficieGanadosJ1
ultimos20SuperficiePerdidosJ1
ultimos30SuperficiePerdidosJ1
ganadosAnioSuperficieJ1
perdidosAnioSuperficieJ1
perdidosAnioAnteriorSuperficieJ1
ganadosSuperficieJ1
perdidosSuperficieJ1
ganadosMesSuperficieJ1
ganadosPeriodoSuperficieJ1
perdidosPeriodoSuperficieJ1
ganadosRondaSuperficieJ1
jugador2
```

```
edadJ2
puntosJ2
posJ2
puntosPasadoJ2
posPasadoJ2
progresionPuntosJ2
progresionPosJ2
AcesJ2
PPSJ2
SGWJ2
BPSJ2
CabezaSerieJ2
RachaJ2
ultimos5GanadosJ2
ultimos5PerdidosJ2
ultimos10GanadosJ2
ultimos30GanadosJ2
ultimos30PerdidosJ2
ganadosAnioJ2
ganadosAnioAnteriorJ2
ganadosJ2
ganadosRondaJ2
ultimos5SuperficieGanadosJ2
ultimos20SuperficieGanadosJ2
ultimos30SuperficieGanadosJ2
ultimos50SuperficiePerdidosJ2
ganadosAnioSuperficieJ2
ganadosAnioAnteriorSuperficieJ2
ganadosMesSuperficieJ2
perdidosMesSuperficieJ2
ganadosPeriodoSuperficieJ2
perdidosPeriodoSuperficieJ2
ganadosRondaSuperficieJ2
perdidosRondaSuperficieJ2
caraAcaraPerdidos
pista
```

6.3.4.3 Algoritmos de clasificación

Se ejecutaron los mismos algoritmos que en los experimentos anteriores realizando validación cruzada de diez subconjuntos de datos en todos ellos.

ZeroR

Como siempre la ejecución de este algoritmo sirvió de referencia para los demás algoritmos de clasificación. Como los ejemplos eran los mismos en todos los subconjuntos de datos y eran los mismos que en el Experimento 1 el resultado de todos ellos debería de ser el mismo.

Se ejecutó el algoritmo con los cuatro subconjuntos de datos y se obtuvo el mismo resultado que en el Experimento 1, es decir, un porcentaje de acierto del 50,11% para todos los subconjuntos.

OneR

Era previsible que este algoritmo también obtuviese el mismo resultado para los cuatro subconjuntos de datos, y que además, fuese el mismo que en el Experimento 1, esto es así porque el algoritmo sólo elige un atributo para clasificar cada ejemplo y se supone que este atributo es el que mejor clasifica los ejemplos. Al haber creado los subconjuntos con algoritmos de selección atributos se preveía que el algoritmo elegido en el Experimento 1 por el algoritmo `OneR` estuviera dentro de todos los subconjuntos creados. Se hizo la comprobación y efectivamente, el atributo elegido en el Experimento 1 por el algoritmo `OneR` era `CabezaDeSerieJ1` el cual aparecía en todos los subconjuntos generados, siendo en aquellos que establecían un ranking de los atributos el primero en todos ellos.

Sólo quedaba por tanto ejecutar el algoritmo con los cuatro subconjuntos y ver que los resultados eran todos iguales y también coincidían con los del Experimento 1. De esta forma se comprobó que todos tenían un porcentaje de acierto del 59,3223 %.

DecisionTable

A partir de la ejecución de este algoritmo se empezó a observar si la selección de atributos había mejorado los resultados con respecto a los obtenidos en el Experimento 1. En este caso, el resultado obtenido en el Experimento 1 con la aplicación del algoritmo `DecisionTable` fue de un 64,02% de acierto.

Se ejecutó por tanto el algoritmo a cada uno de los subconjuntos y sin embargo, no se obtuvo mejora en ninguno de los casos con respecto al Experimento 1. Los porcentajes de acierto de los subconjuntos `A`, `B`, `C` y `D` respectivamente fueron del 63,76% para los dos primeros y del 63,84% y 63,87% para los dos últimos. Los resultados no empeoran demasiado pero si empezaron a descartar que la selección de atributos fuese a ser un paso clave para la mejora de resultados.

NaiveBayes

El resultado obtenido por este algoritmo fue de los que peor porcentaje de acierto cosechó en el Experimento 1, en este apartado se podrá observar si se supera dicho porcentaje, que en este caso fue del 61,23% de acierto.

La prueba fue satisfactoria obteniendo los mejores resultados en los subconjuntos de datos `A`, `B` y `C` todos superando el 64% de acierto, sin embargo el subconjunto `D` únicamente consiguió el 62,82% de acierto. Por tanto, se pudo concluir que el algoritmo funcionaba mejor con pocos atributos pues el subconjunto `D` tenía casi tres veces más atributos que los otros subconjuntos y el conjunto de datos total tenía más del doble que el subconjunto `D` siendo el que peor resultados obtuvo.

Sin embargo, este algoritmo seguía sin mejorar el resultado ofrecido por otros algoritmos en el Experimento 1, aunque en realidad si se aproximó mucho al mejor de ellos.

C4.5

El algoritmo `C4.5` obtuvo distintos resultados en cada uno de los subconjuntos. El que peor resultado obtuvo fue el subconjunto `D` que incluso empeoró el resultado del algoritmo en el Experimento 1. Los otros tres subconjuntos `A`, `B` y `C` mejoraron los resultados del Experimento 1 pero apenas aumentroan un 1,06% el porcentaje de acierto.

Se puede observar que todos los resultados variaron a pesar de que había subconjuntos de datos muy similares, esto fue debido a que los árboles generados en cada uno de los casos eran completamente distintos. Para poder ver estas diferencias se muestra el gráfico de la Tabla 21 en el que se detallan el tamaño y el número de hojas de cada uno de los árboles.

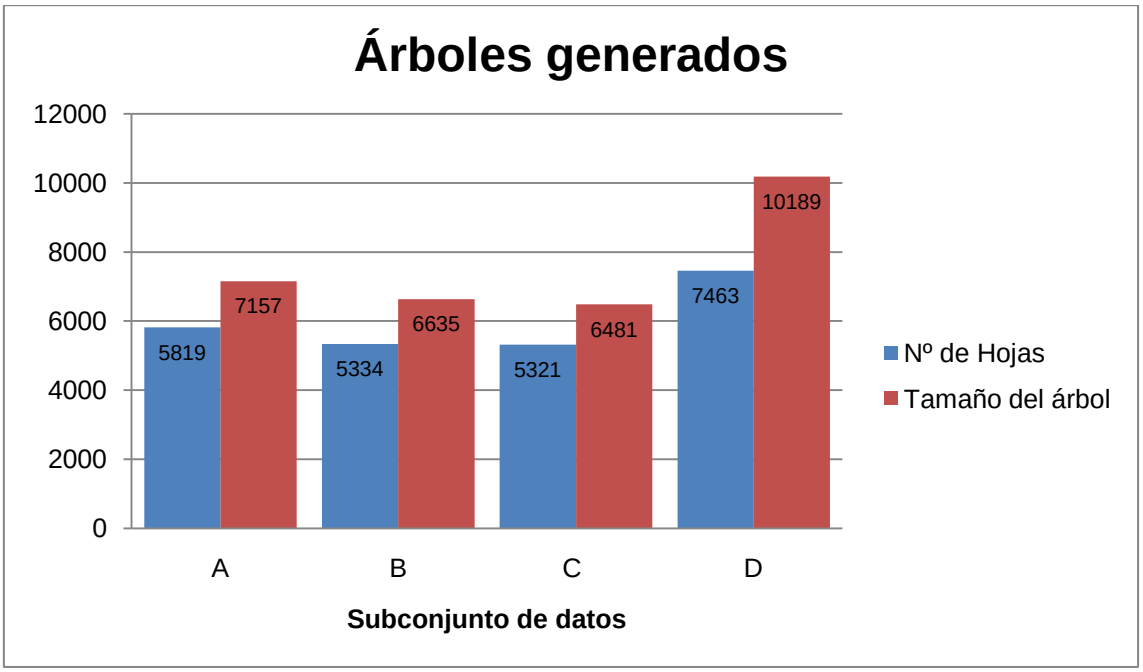


Tabla 21: Árboles generados con el algoritmo `C4.5` en el experimento 4.

IB1

Los resultados obtenidos en este experimento por este algoritmo sirvieron para descartar su ejecución en futuros experimentos debido a sus pésimos resultados. En este caso se pudo observar que el algoritmo obtenía mejores resultados con menos atributos pues con los subconjuntos `A`, `B` y `C` consiguió un 57% de acierto muy cerca del 58% en algún caso. Sin embargo, tanto en el Experimento 1, con un 52,71%, como en el subconjunto `D` con un 53, 28% fueron superados incluso por el algoritmo `OneR`, el más simple de todos (sin tener en cuenta el `ZeroR`), que obtenía un porcentaje del 59,32%.

AdaBoostM1

Los resultados obtenidos en los distintos subconjuntos fueron muy similares entre todos ellos y también en comparación con los obtenidos en el Experimento 1. En este caso el peor resultado fue obtenido en el subconjunto `D` con un porcentaje de acierto del 62,06%, mientras que el mejor resultado fue obtenido por el subconjunto de datos `B` con el 63,18% de las instancias correctamente clasificadas.

No se consiguió mejorar los resultados obtenidos con otros algoritmos, y sin embargo, la duración del proceso de ejecución del algoritmo fue mucho mayor que en otros casos.

Como ya se describió en experimentos anteriores el algoritmo genera diez árboles de decisión. A continuación, se muestra el gráfico de la Tabla 22 con la media de hojas y del tamaño de los árboles en cada uno de los subconjuntos, en él se puede observar que el subconjunto con un mayor número de atributos generó árboles mucho mayores.

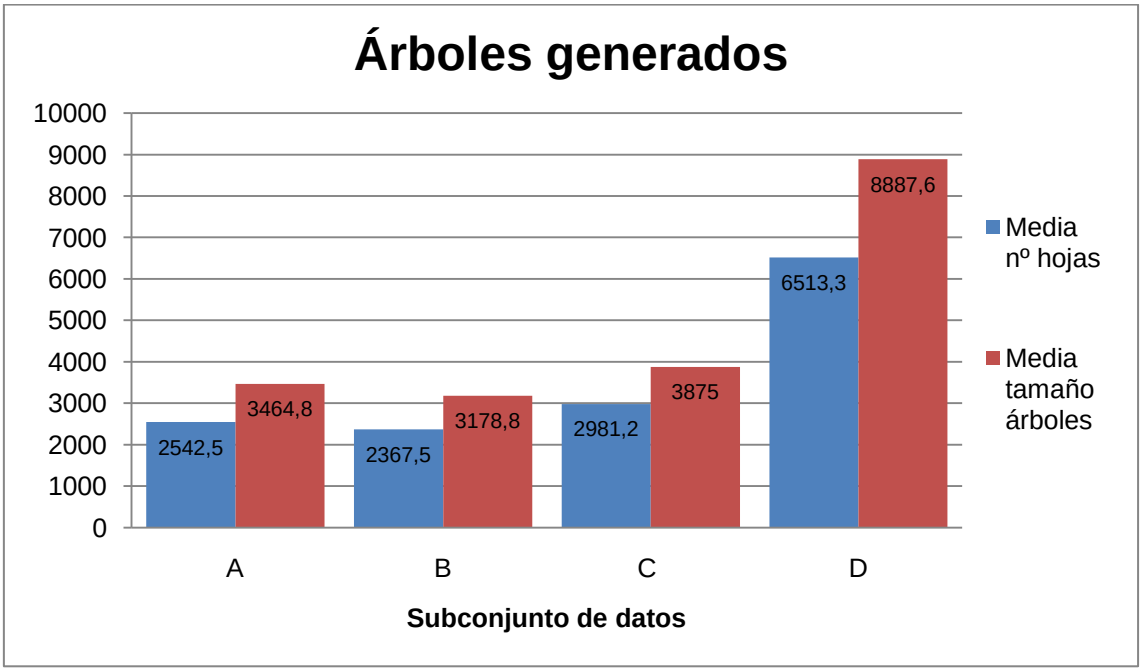


Tabla 22: Árboles generados con el algoritmo `AdaBoostM1` en el experimento 4.

Bagging

Ninguno de los resultados obtenidos al ejecutar el algoritmo con los subconjuntos de datos creados mejoró el resultado del Experimento 1. En este caso todos los resultados fueron similares siendo el resultado que más bajó su porcentaje de acierto con respecto al Experimento 1 el obtenido por el subconjunto `C` que bajó un 1,09%.

A continuación, se ofrece el gráfico de la Tabla 23 que es similar al anterior y es útil para comparar el tamaño de los árboles generados. Es curioso el resultado al ser comparado con el algoritmo `AdaBoostM1`. En este caso los árboles más grandes fueron generados para los subconjuntos con menor número de atributos, todo lo contrario a lo ocurrido en el algoritmo `AdaBoostM1` en donde a mayor número de atributos mayor tamaño tenían los árboles generados.

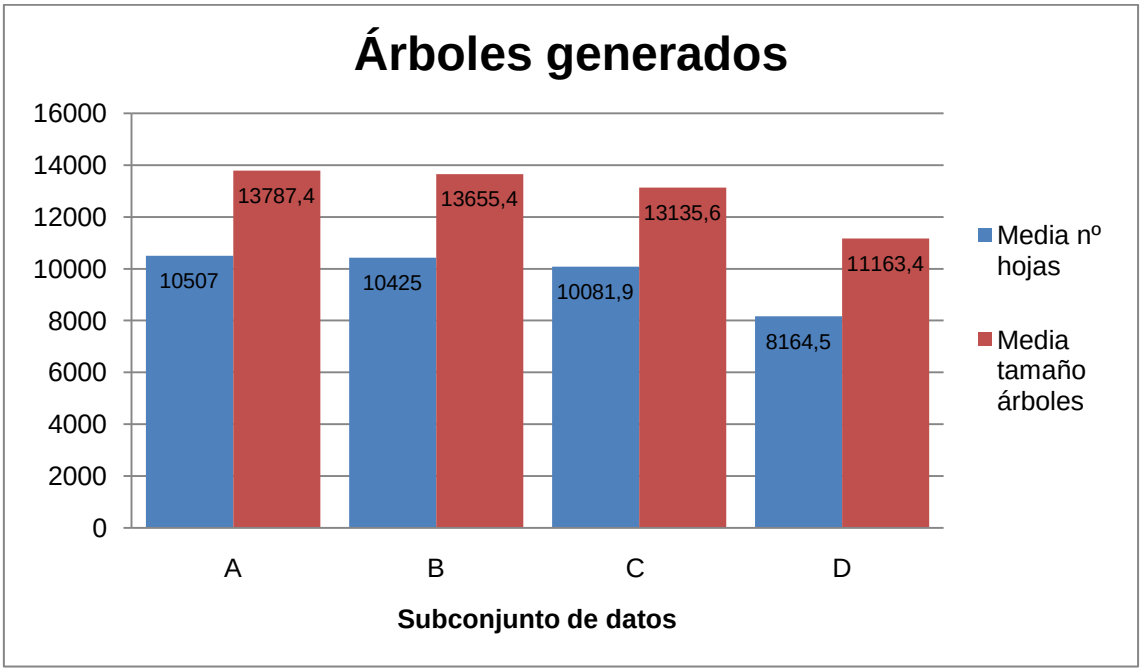


Tabla 23: Árboles generados con el algoritmo `Bagging` en el experimento 4.

6.3.4.4 Resultados del experimento

Una vez ejecutados todos los algoritmos en todos los subconjuntos se comparan los distintos resultados en el gráfico de la Tabla 24. Para ello se ofrece un gráfico con los resultados de todos los subconjuntos. También han sido incorporados los resultados obtenidos en el Experimento 1. Los resultados de los algoritmos `ZeroR` y `OneR` no se muestran en el gráfico pues fueron idénticos para todos los experimentos como ya se describió en puntos anteriores.

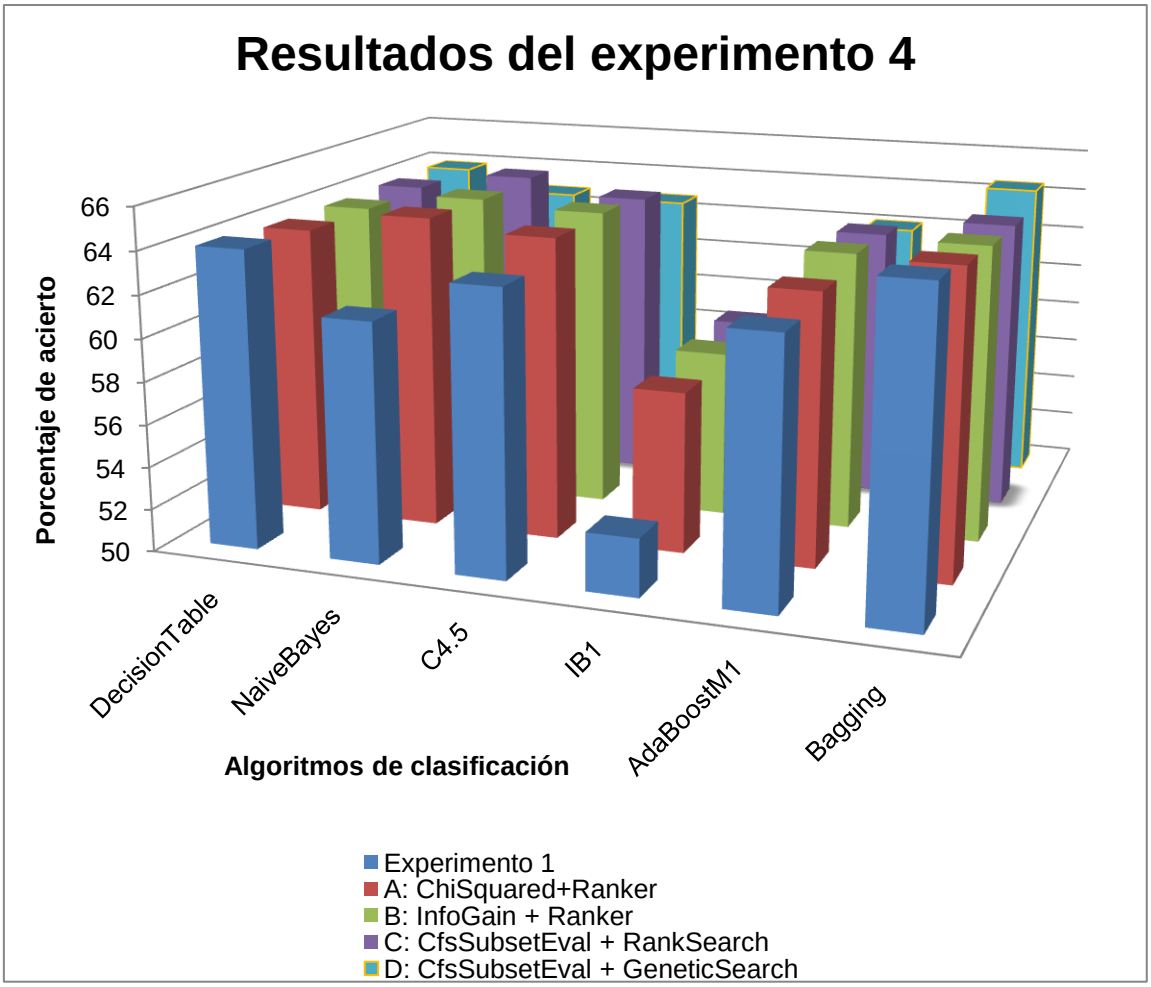


Tabla 24: Resumen de resultados del experimento 4.

El mejor resultado fue obtenido por el Experimento 1 con el algoritmo `Bagging`, con el cual se obtuvo un porcentaje de acierto del 64,88%. El algoritmo `DecisionTable` también obtuvo mejores resultados en el Experimento 1 que en cualquiera de los subconjuntos del Experimento 4. Sin embargo, los cuatro algoritmos restantes sí que mejoraron sus resultados con respecto al Experimento 1 y en dos de los casos, `NaiveBayes` e `IB1`, de forma notable con mejoras de más de un 3% y un 5% respectivamente.

Con la elaboración del experimento no se puede concluir de forma general que se obtengan mejores resultados con la eliminación de algunos. Sin embargo, si se hiciera la media de los resultados obtenidos sí que se podría decir que los resultados habrían mejorado en todos los casos los del Experimento 1 pues en éste la media del porcentaje de acierto obtenida fue del 61,39% mientras que en el Experimento 4 las medias fueron de 62,87%, 62,94%, 62,87% y 61.54% para los subconjuntos de datos `A`, `B`, `C` y `D` respectivamente.

Por tanto de forma general se puede decir que el subconjunto que obtuvo mejores resultados fue el subconjunto `B`.

6.3.5: Experimento 5: Estadísticas completas y apuestas

En los experimentos anteriores se utilizaron estadísticas de los jugadores que disputaban un partido y por otro lado, los datos de apuestas realizadas en *Betfair*. Este experimento sirvió de unión entre los experimentos anteriores y de esta forma obtener un resultado apoyándose en la combinación de estadísticas y datos de apuestas. Además, cabe resaltar que a partir de este experimento sólo fueron contempladas las apuestas realizadas antes del inicio de los partidos, es decir, el experimento estaba dirigido para intentar predecir el resultado del partido antes del comienzo. Al tener en cuenta sólo las cuotas antes del inicio de los partidos, para hacer una simulación real de una inversión en el mercado sólo se podía apostar a la última cuota a la que se haya apostado justo antes del inicio de cada partido. Esto tenía la desventaja de que no se podía apostar en los picos de cuota, pero a su vez tenía la ventaja de que los datos de las cuotas aportaban información importante de las probabilidades de victoria de cada uno de los jugadores. Con la evaluación de este experimento se pudo observar de qué forma se obtenían mejores resultados.

6.3.5.1 Conjunto de datos

El primer paso para llevar a cabo el experimento fue la formación de los conjuntos de datos. En el fichero de datos del Experimento 1 se disponía de 57.400 partidos. Sin embargo, no de todos estos partidos se disponían datos de apuestas, ya que muchos de ellos pertenecían a torneos de nivel nacional los cuales no eran contemplados por la casa de apuestas *Betfair*.

Las tablas principales de la base de datos para formar el fichero de datos de este experimento fueron la tabla `Betfair` y la tabla `Partidos`. Se recorrió la tabla `Partidos` consultando todos los partidos que estaban relacionados con alguna línea de la tabla de datos `Betfair`, es decir, todos los partidos en los que se habían realizado apuestas.

En total se dispuso de 12.673 partidos en los que se habían realizado apuestas. Este descenso del número de partidos fue debido a que en la tabla `Partidos` se disponía de partidos jugados desde el 2002 mientras que los primeros datos de apuestas obtenidos databan de junio del 2004 Aademás, como se describió anteriormente, en la tabla `Partidos` se podían encontrar partidos entre jugadores amateur o semiprofesionales los cuales no eran ofrecidos para apostar en *Betfair*.

Cada uno de los 12.673 ejemplos disponibles tenía 161 atributos, estos atributos estaban compuestos por todos los atributos del Experimento 1 (a excepción del atributo `premiosTorneo` que ocasionaba muchos problemas al ser de tipo cadena de caracteres), y por ocho atributos más relacionados con las cuotas a favor de cada uno de los jugadores. Para cada uno de los jugadores se registraron las cuotas mínima, máxima, media y la última cuota de la última apuesta realizada antes del inicio del

partido. Todos los atributos de este experimento se muestran en el *Anexo E: Atributos del Experimento 5*.

En este experimento también se creó otro subconjunto de datos con el algoritmo de selección de atributos que mejor funcionó en el Experimento 4, es decir, el algoritmo `InfoGainAttributeEval`, y además, se compararon los resultados obtenidos con y sin cuotas, creando dos nuevos subconjuntos (sin cuotas) a partir de los anteriores. Por tanto, se trabajó a lo largo del experimento con cuatro conjuntos de datos:

A. **Estadísticas completas con cuotas**: Este subconjunto estaba formado por todos los atributos incluidos en la Tabla 90 (ver *Anexo E: Atributos del Experimento 5*).

B. **Estadísticas completas sin cuotas**: Formado por el conjunto anterior sin los ocho atributos con información pertenecientes a las cuotas.

C. **Mejores 20 atributos de estadísticas completas con cuotas**: Formado por los mejores 20 atributos obtenidos con el algoritmo de selección de atributos `InfoGainAttributeEval` de los datos de estadísticas completas y los 8 datos de las cuotas.

D. **Mejores 20 atributos de estadísticas completas sin cuotas**: Formado por los mejores 20 atributos obtenidos con el algoritmo de selección de atributos `InfoGainAttributeEval` de los datos de estadísticas completas.

6.3.5.4 Algoritmos de selección de atributos.

En este apartado se detallan qué dos subconjuntos de datos se utilizaron para crear los ficheros de datos `C` y `D`. Como ya se señaló el algoritmo de selección de atributos utilizado fue con el que mejores resultados medios se obtuvieron en el Experimento 4, es decir, el `InfoGainAttributeEval` en combinación con el método de búsqueda `Ranker`.

Los veinte mejores atributos obtenidos al aplicar el algoritmo fueron los siguientes:

```
Ranked attributes:

 0.051006      4 posJ1
 0.050437     26 CabezaSerieJ1
 0.047109      3 puntosJ1
 0.046978     76 posJ2
 0.045597     98 CabezaSerieJ2
 0.045412     75 puntosJ2
 0.037706      5 puntosPasadoJ1
 0.037442      6 posPasadoJ1
 0.035572     78 posPasadoJ2
```

```
 0.03469      77 puntosPasadoJ2
 0.030101      1 jugador1
 0.029112     25 PremiosJ1
 0.027672      9 TMWJ1
 0.027082     81 TMWJ2
 0.025938     97 PremiosJ2
 0.024175     37 ultimos50PerdidosJ1
 0.024175     36 ultimos50GanadosJ1
 0.023682    114 ganadosJ2
 0.023344     85 MFJ2
 0.023038     42 ganadosJ1
```

Con estos atributos, más la clase y los ocho atributos de las cuotas se formó el subconjunto de datos `C`. Si al subconjunto de datos `C` se le quitaban los 8 atributos de las cuotas se formaba el subconjunto de datos `D`.

Cabe destacar la diferencia existente con el resultado obtenido por el mismo algoritmo en el Experimento 4 en el cual ninguno de los atributos `PremiosJ1, TMWJ1, TMWJ2, PremiosJ2 y MFJ2` estaba entre los mejores. Esta diferencia fue debida a la eliminación de muchos de los ejemplos que formaban parte del conjunto de datos inicial en el Experimento 4. Por ejemplo, la inclusión de los atributos con la información de los premios conseguidos por cada uno de los jugadores pudo ser debida a que el conjunto inicial de datos de este experimento sólo estaba formado por partidos de tenistas profesionales, por lo tanto, el volumen de dinero ganado por cada uno de los jugadores estaba directamente relacionado con el resultado conseguido en los torneos disputados.

6.3.5.3 Algoritmos de clasificación

En este experimento no se ejecutó el algoritmo `IB1` por los pobres resultados obtenidos en experimentos anteriores y por el tiempo de ejecución requerido por el algoritmo. Como en la mayoría de los experimentos anteriores se realizó validación cruzada de diez subconjuntos. Además, se ejecutaron dos nuevos algoritmos con los que, hasta ahora, no se había experimentado, estos fueron los algoritmos `REPTree` y `ConjuctiveRule`. También se probaron los resultados obtenidos por el algoritmo `Bagging` en combinación con el árbol de decisión `REPTree`. Además, se dejaron de ejecutar los algoritmos `ZeroR` y `OneR` pues ya se tenían como referencia de resultados a mejorar los obtenidos en experimentos anteriores.

DecisionTable

Los resultados obtenidos en los subconjuntos de datos `A` y `C`, es decir los que tenían datos de cuotas fueron iguales con un porcentaje de acierto del 70,48%. Sin embargo, los resultados obtenidos en los subconjuntos sin datos de las cuotas empeoraron más de un 5%. En este caso el subconjunto `B` tuvo un 64,83% de acierto mientras que el `D` consiguió un 65,05%.

Por tanto, en este algoritmo quedó reflejada la importancia de la información que llevaban consigo los valores de las cuotas, que al ser incluidos mejoraron claramente los porcentajes de acierto.

NaiveBayes

En este caso la diferencia entre los subconjuntos con cuotas y sin ellas no era tan clara como en el algoritmo anterior. Aún así, los resultados volvieron a ser mejores en los subconjuntos con cuotas. En este caso, el mejor resultado se obtuvo con el subconjunto `C` con el que se consiguió un porcentaje de acierto del 67,13%.

Los subconjuntos `A`, `B` y `C` obtuvieron unos resultados del 66,88%, 65,66% y 65,32% de acierto respectivamente.

C4.5

Los resultados obtenidos con este algoritmo sorprendieron por los cambios que presentaron de unos subconjuntos a otros. En este caso, como ocurría con el algoritmo `NaiveBayes`, el mejor de los resultados fue obtenido por el subconjunto de datos `C` con un porcentaje de acierto del 68,84%. El siguiente en el ranking de resultados con este algoritmo fue el otro subconjunto con datos con cuotas, es decir, el subconjunto `A` que obtuvo un acierto del 66,15%.

Los resultados obtenidos por los subconjuntos `B` y `D` fueron del 61.92% y 64.25% de acierto respectivamente.

AdaBoostM1

Este algoritmo no obtuvo buenos resultados pues, en ninguno de los casos fue superado el 66% de acierto que sí había sido superado por los tres algoritmos anteriores.

Los resultados obtenidos fueron del 65,89%, 62,78%, 64,94% y 61,58% para los subconjuntos `A, B, C y D` respectivamente. Nuevamente se pudo observar como los mejores resultados se consiguieron en los dos subconjuntos con los datos de las cuotas.

Bagging

Se experimentó con dos posibles configuraciones del algoritmo. La primera de ellas fue la llevada a cabo en los experimentos anteriores, es decir, con el algoritmo `C4.5` y la segunda la establecida por defecto por el software utilizado (Weka), es decir, con el algoritmo `RepTree`.

- C4.5

Volvió a ocurrir lo mismo que ya ocurría con la ejecución de otros algoritmos como `DecisionTable` o `NaiveBayes`, en donde los resultados eran similares entre los subconjuntos con cuotas por una parte, y los subconjuntos sin cuotas por otra. En este caso los subconjuntos `A` y `C` obtuvieron un 68,46% y un 68,47% de acierto respectivamente mientras que los subconjuntos `B` y `D` tuvieron unos resultados del 65,85% y del 64,39% de acierto. Por tanto, los resultados obtenidos en los subconjuntos con cuotas volvieron a obtener resultados mucho mejores que los subconjuntos que carecían de la información proporcionada por los atributos de las cuotas.

- RepTree

Al ser la primera vez que se ejecutaba el algoritmo se detallan a continuación los parámetros del algoritmo `RepTree` (los del algoritmo `Bagging` fueron los mismos) que como en la mayoría de los casos eran los establecidos por defecto en el software utilizado, es decir, en Weka. El primer parámetro

configurable es la profundidad máxima "maxDepth = -1", en este caso el valor por defecto es -1, es decir, sin restricciones, lo cual indica que los árboles generados por el algoritmo pueden alcanzar cualquier profundidad. El siguiente parámetro "minNum = 2.0" establece el peso mínimo total de las instancias en las hojas. Con el parámetro "minVarianceProp = 0.0010" se establece el porcentaje mínimo de la varianza en todos los datos que se necesita para estar presentes en un nodo con el fin de dividir los árboles de regresión generados. El siguiente parámetro "noPruning = False" indica que en este caso la realización de la poda está habilitada para la generación de los árboles. Con el parámetro "numFolds = 3" se determina la cantidad de los datos que son usados para la poda, siempre es uno de los subconjuntos siendo el resto de los subconjuntos en los que se dividen los datos usados para el desarrollo de las reglas. Por último, el parámetro "seed = 1" indica la semilla usada para realizar la asignación aleatoria de los datos.

Los resultados obtenidos con esta nueva combinación en este caso mejoraron ligeramente el resultado en los subconjuntos con cuotas, sin embargo, empeoraron ligeramente los obtenidos en los subconjuntos sin cuotas. De esta forma, los porcentajes de acierto para los subconjuntos con cuotas A y C fueron del 69,38% y 69,07% respectivamente, mientras que en los subconjuntos sin cuotas B y D fueron del 65,48% y 64,24% de acierto.

REPTree

Los parámetros de configuración fueron los mismos que los utilizados en el clasificador RepTree utilizado en combinación con el algoritmo Bagging.

Los resultados mejoraron al otro árbol de decisión contemplado en el experimento, es decir al C4.5, y se acercaron a los mejores obtenidos hasta el momento que eran los conseguidos por el algoritmo DecisionTable. En este caso los subconjuntos con cuotas, A y C, obtuvieron unos porcentajes de acierto del 69,26% y 68,77%, mientras que los subconjuntos sin cuotas B y D sólo consiguieron acertar en el 63,84% y 63,3% de los casos. Volvió a quedar claro que la inclusión de las cuotas mejoraba considerablemente los resultados en cuanto al porcentaje de acierto, en el Capítulo 7 se comprobará si también esto se traducía en beneficios económicos.

ConjuctiveRule

Al ser la primera vez que se ejecutó este algoritmo se procede a la descripción de la configuración de los parámetros. Con el primer parámetro configurable "exclusive = false" se indica que no se considerarán expresiones exclusivas para la división de atributos nominales. El parámetro "folds = 3" determina la cantidad de los datos que serán usados para la poda, que será siempre de uno de los subconjuntos y el restos de los subconjuntos en los que se dividen los datos serán usados para el desarrollo de las reglas. El siguiente parámetro "minNum = 2.0" establece el peso mínimo total de las instancias en las hojas. A continuación, "numAntds = -1" establece el número de antecedentes permitidos en la regla si es

usada la pre poda. Si este valor es distinto de -1, entonces, se usará pre poda, en caso contrario la regla usa el error reducido en la poda. Por último, el parámetro "seed = 1" indica la semilla usada para realizar la asignación aleatoria de los datos.

Los resultados obtenidos con la aplicación del algoritmo fueron los que más diferenciaron los subconjuntos con cuotas de los sin cuotas, pues, se obtuvieron diferencias de más del 10% entre ellos. En este caso, los resultados de los subconjuntos con cuotas A y C fueron del 70,24% y 70,21% de acierto respectivamente mientras que los resultados de los subconjuntos sin cuotas B y D tuvieron únicamente un acierto del 59,77% y 60,21%.

6.3.5.4 Resultados del experimento

A continuación se muestran algunos gráficos que servirán para comparar los resultados obtenidos con los distintos subconjuntos de datos y algoritmos aplicados.

El gráfico de la Tabla 25 es válido para comprobar en líneas generales qué subconjunto se comportó mejor. En este caso el subconjunto sobre el que de media se obtuvieron mejores resultados fue el subconjunto C seguido muy de cerca por el subconjunto A.

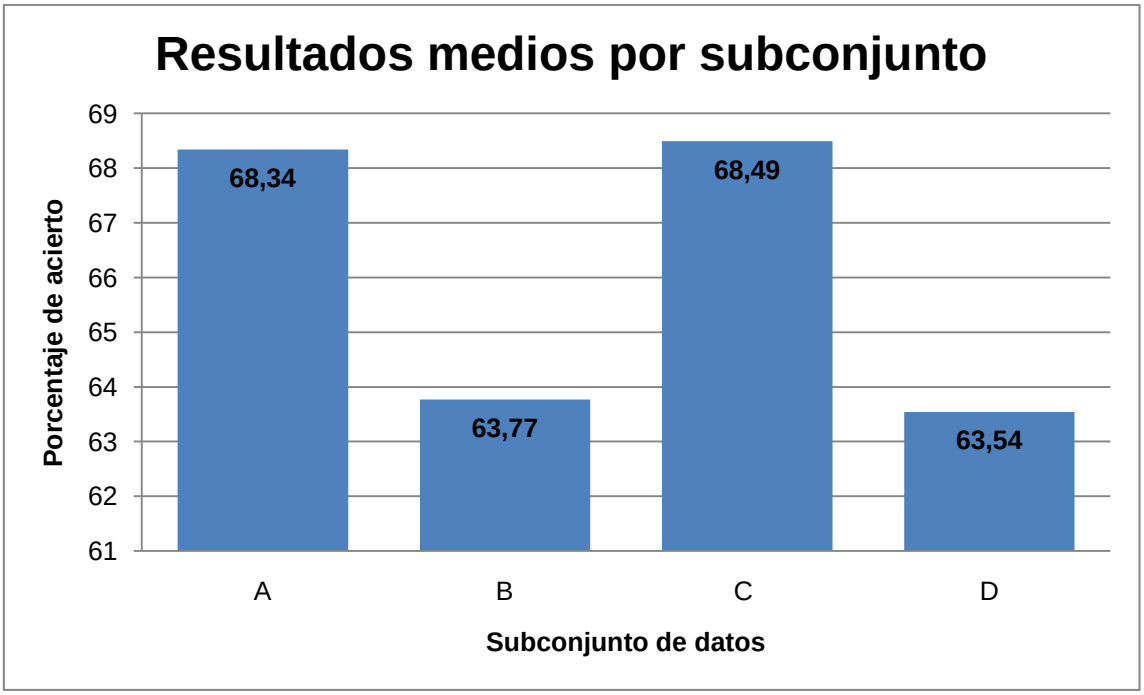


Tabla 25: Comparación de resultados por subconjuntos del experimento 5.

Es decir, los dos subconjuntos que mejores resultados obtuvieron fueron los que contaban con atributos con información de las cuotas de los distintos partidos. Esto se puede ver muy claramente en el gráfico de la Tabla 26 en el que se puede apreciar que como media fueron casi un 5% mejores los resultados obtenidos en los subconjuntos con datos de las cuotas.

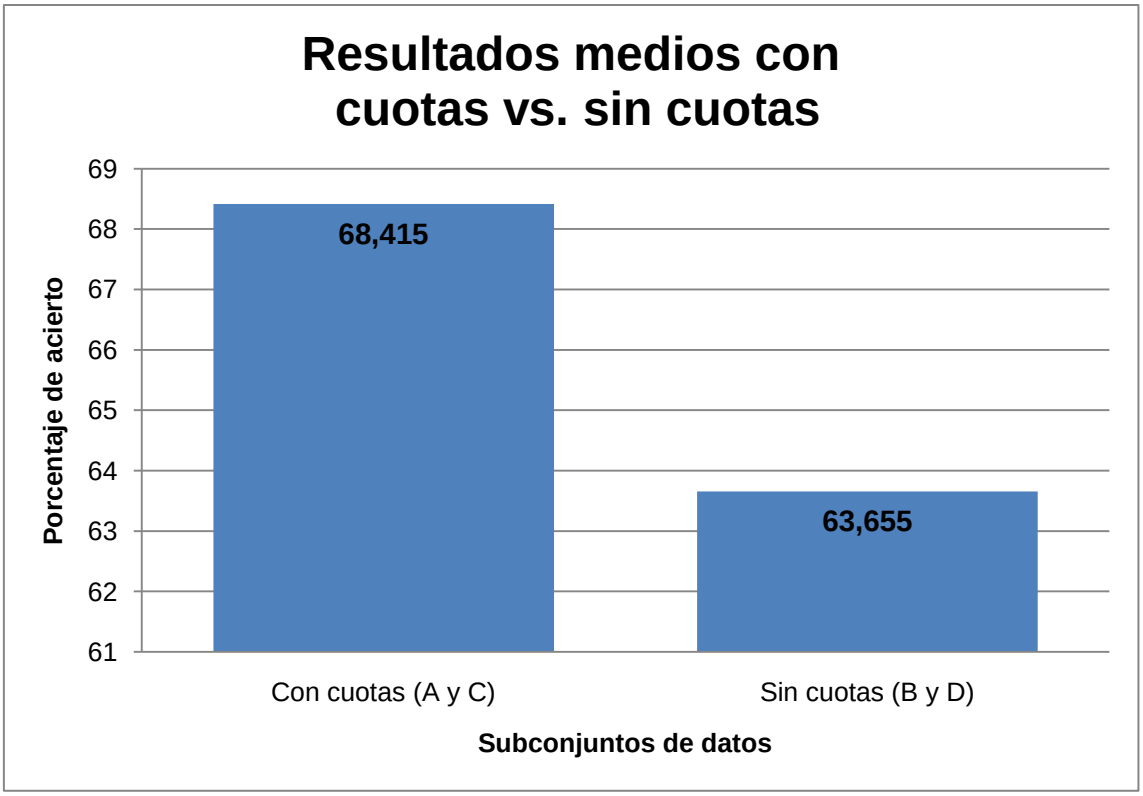


Tabla 26: Comparación de resultados de datos con cuotas vs. sin cuotas del experimento 5.

En el gráfico de la Tabla 27 se pueden comparar de forma visual todos los resultados. De esta forma se aprecia que, a excepción del algoritmo `C4.5`, los resultados iban a la par en los subconjuntos `A` y `C`, es decir, los subconjuntos con datos de cuotas, y los subconjuntos `B` y `D`, en los que no había datos de cuotas. Además, se aprecia una ligera mejoría de los subconjuntos `A` y `B` con respecto a los subconjuntos `C` y `D` es decir, que se obtuvieron unos resultados ligeramente mejores en los subconjuntos en los que no se había realizado selección de atributos, también la clara excepción a esto se presentó en el algoritmo `C4.5` donde los subconjuntos `A` y `B` fueron superados por `C` y `D` respectivamente.

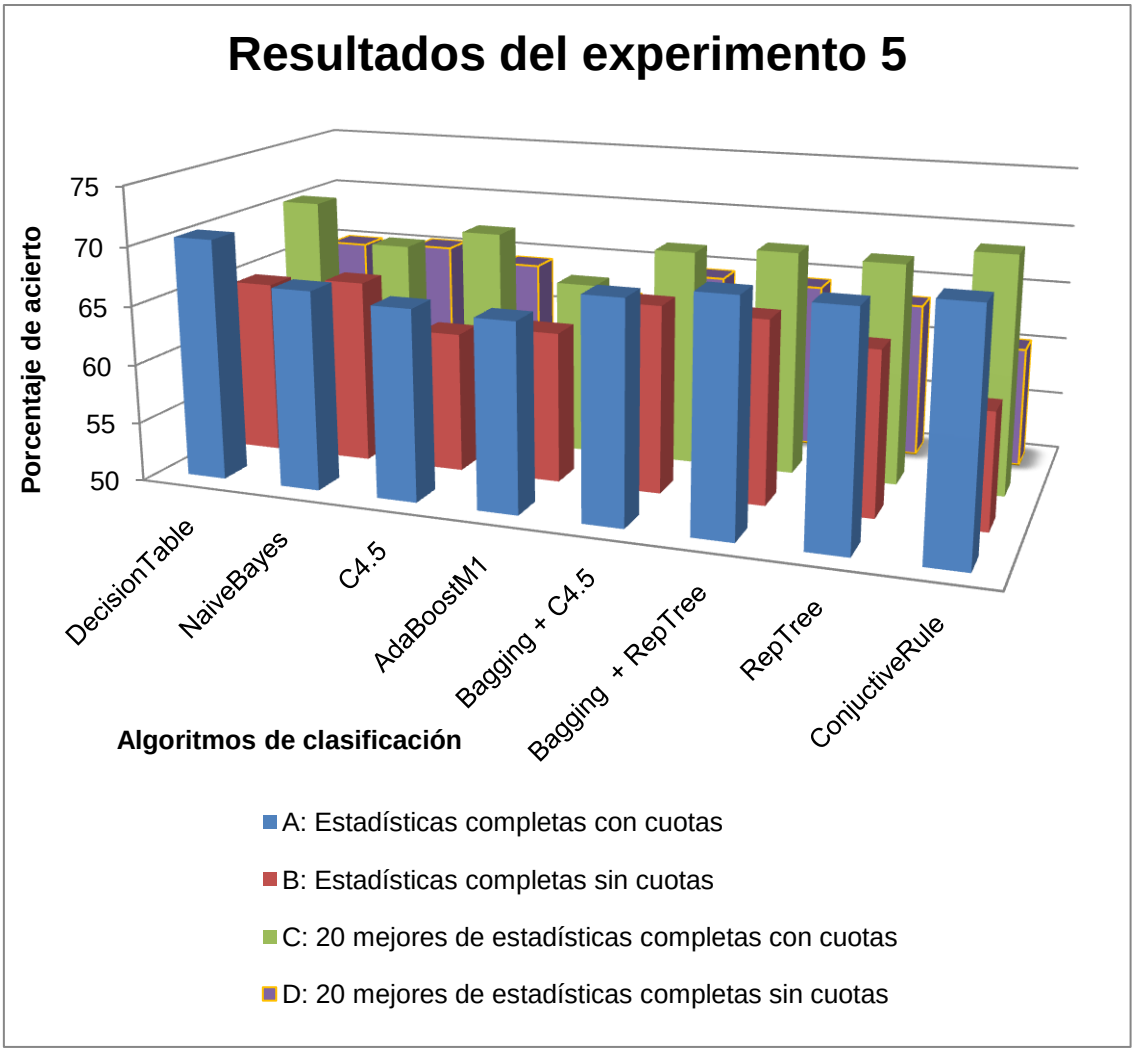


Tabla 27: Resumen de resultados del experimento 5.

6.3.6: Experimento 6: Estadísticas enfrentadas y apuestas

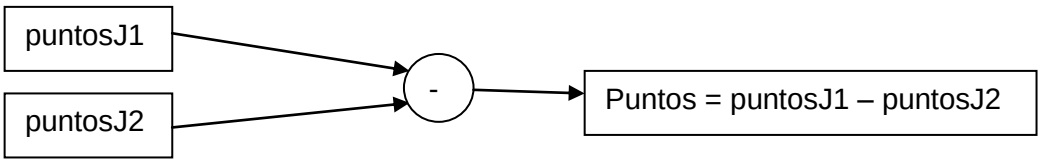
En este experimento se realizaron los mismos pasos que en el experimento anterior, únicamente se diferenció del Experimento 5 en el fichero de datos. A lo largo del proyecto se intentó con las herramientas ofrecidas por Weka, en este caso con el apartado de algoritmos de asociación, hacer reglas que relacionasen los atributos entre sí para determinar la clase, estas reglas serían del tipo:

```
Si puntosJ1 > puntosJ2 => ganador = 1
```

Sin embargo, la aplicación de estos algoritmos no tuvo éxito y no se consiguieron buenos resultados. Por lo tanto, se buscó una alternativa que dio como consecuencia la realización de este experimento.

6.3.6.1 Conjunto de datos

El conjunto de datos de este experimento se generó a partir del conjunto de datos utilizado en el Experimento 5. En este conjunto la mayoría de los atributos eran estadísticas de ambos jugadores, en base a esto se decidió enfrentar cada una de las estadísticas de un jugador con su misma estadística del otro jugador para que formasen un único atributo, por ejemplo:



De esta forma, se redujo el número de atributos en casi la mitad, por tanto cada uno de los ejemplos que componían el fichero de datos estaba formado por 79 atributos detallados en el *Anexo F: Atributos del Experimento 6*.

Como en el Experimento 5, en este caso también se formaron cuatro subconjuntos de datos para llevar a cabo distintas pruebas y comparar resultados. Los subconjuntos con los que se trabajó a lo largo del experimento fueron los siguientes:

A. **Estadísticas enfrentadas completas con cuotas**: Este subconjunto estaba formado por todos los atributos incluidos en la Tabla 91 (ver *Anexo F: Atributos del Experimento 6*).

B. **Estadísticas enfrentadas completas sin cuotas**: Formado por el conjunto anterior sin los ocho atributos con información pertenecientes a las cuotas.

C. **Mejores 20 atributos de las estadísticas enfrentadas completas con cuotas**: Formado por los mejores 20 atributos obtenidos con el algoritmo de selección de atributos `InfoGainAttributeEval` de los datos de estadísticas completas y los 8 datos de las cuotas.

D. **Mejores 20 atributos de las estadísticas enfrentadas completas sin cuotas**: Formado por los mejores 20 atributos obtenidos con el algoritmo de selección de atributos `InfoGainAttributeEval` de los datos de estadísticas completas.

6.3.6.2 Algoritmos de selección de atributos

En este apartado se detallan los dos subconjuntos de datos obtenidos al aplicar la selección de atributos para crear los ficheros de datos `C` y `D`. Como ya ha sido señalado el algoritmo de selección de atributos utilizado fue con el que mejores resultados medios se obtuvieron en el Experimento 4, es decir, el `InfoGainAttributeEval` en combinación con el método de búsqueda `Ranker`.

Los veinte mejores atributos obtenidos al aplicar el algoritmo fueron los siguientes:

```
Ranked attributes:

 0.110674    10 puntos
 0.097871    11 pos
 0.081954    12 puntosPasado
 0.065482    13 posPasado
 0.061836    43 ganados
 0.055609    16 TMW
 0.051255    32 Premios
 0.050437     7 CabezaSerieJ1
 0.047252    38 ultimos50Ganados
 0.045597     8 CabezaSerieJ2
 0.043229    37 ultimos30Ganados
 0.042563    20 MF
 0.041903    45 ganadosMes
 0.039452    49 ganadosRonda
 0.039076    36 ultimos20Ganados
 0.038251    41 ganadosAnioAnterior
 0.037517    61 ganadosSuperficie
 0.037028    39 ganadosAnio
 0.036166    63 ganadosMesSuperficie
 0.035963    18 TBW
```

Con estos atributos, más la clase y los ocho atributos de las cuotas se formó el subconjunto de datos `C`. Quitándole al subconjunto de datos `C` los 8 atributos de las cuotas se formó el subconjunto de datos `D`.

6.3.6.3 Algoritmos de clasificación

Se ejecutaron exactamente los mismos algoritmos y con los mismos parámetros que en el Experimento 5. Los resultados fueron similares en cuanto a que los subconjuntos `A` y `C` iban a la par en resultados al igual que los subconjuntos `B` y `D`, es decir, los que llevaban cuotas y los que no llevaban cuotas entre sus atributos.

DecisionTable

Los subconjuntos `A` y `C`, es decir, los subconjuntos que tenían atributos con información de las cuotas obtuvieron exactamente los mismos resultados que en el experimento 5, esto fue debido a que las reglas generadas por el algoritmo se basaban en los valores de los atributos de las cuotas, los cuales eran iguales en ambos experimentos. Por tanto, el resultado en estos dos subconjuntos volvió a ser del 70,48% de acierto.

Por lo que refiere a los subconjuntos `B` y `D` en este caso ambos presentaron resultados similares con un porcentaje de acierto del 66,16% y 66,13% respectivamente. Además, los resultados mejoraron más de un 1% respecto a los obtenidos en el Experimento 5.

También, cabe resaltar que la selección de atributos en este caso no aportó resultados positivos.

NaiveBayes

Este algoritmo es el único junto con el árbol de decisión `RepTree` que mejoró todos los resultados en relación al Experimento 5. Los resultados volvieron a ser similares entre subconjuntos con cuotas y sin ellas pero además entre ellos apenas hubo diferencias del 1%.

Los resultados obtenidos fueron del 67,52% y 67,77% para los subconjuntos con cuotas `A` y `C`, mientras que los subconjuntos `B` y `D` tuvieron unos porcentajes de acierto del 66,73% y 66,67%.

Este es el único algoritmo que presentó resultados similares para todos los subconjuntos siendo la máxima diferencia entre dos subconjuntos de sólo un 1,1%.

C4.5

Si se recuerdan los resultados obtenidos por este algoritmo en el Experimento 5, se observa que era el único caso en el que los subconjuntos `C` y `D` se imponían claramente a los subconjuntos `A` y `B`, es decir, la reducción de los conjuntos de datos mediante algoritmos de selección de atributos obtenía buenos resultados. En este caso volvió a ocurrir exactamente lo mismo, los resultados de los subconjuntos con menos atributos se impusieron a los subconjuntos con todos los atributos y como en todos los experimentos los subconjuntos con cuotas se impusieron a los subconjuntos que no tenían atributos con información de las mismas.

Por tanto, los resultados fueron los siguientes: los subconjuntos `A` y `C` obtuvieron un porcentaje de acierto del 66,10% y 68,67% mientras que los subconjuntos `B` y `D` tuvieron el 63,08% y 63,95% de acierto respectivamente.

AdaBoostM1

Como en el experimento anterior el algoritmo no obtuvo buenos resultados, siendo mejorado por el resto de algoritmos. Nuevamente ningún subconjunto de datos superó el 66% de acierto.

Los resultados obtenidos fueron del 65,21%, 62,89%, 64,08% y 61,60% para los subconjuntos `A, B, C y D` respectivamente. Nuevamente se pudo observar como los mejores resultados se consiguieron en los dos subconjuntos con los datos de las cuotas.

Bagging

Se experimentó otra vez con dos configuraciones del algoritmo como ya se hizo en el Experimento 5.

- *C4.5*

  Los resultados fueron muy similares a los obtenidos en el mismo apartado del Experimento 5 y siguieron la tendencia de la mayoría de los algoritmos, es decir los subconjuntos con cuotas obtenían mejores resultados que los sin cuotas.

  Destacar que el mejor resultado fue conseguido por un subconjunto al que se le había aplicado la selección de atributos, el subconjunto `C` que obtuvo un porcentaje de acierto del 68,62%.. Por otra parte, los subconjuntos `A, B y D` obtuvieron unos porcentajes de acierto del 68,13%, 66,14% y 64,62% respectivamente

- RepTree

Los resultados obtenidos con esta combinación mejoraron en todos los casos los obtenidos por la combinación `Bagging + C4.5` siguiendo su mismo orden de resultados en lo que a los subconjuntos se refiere. En este caso los porcentajes de acierto conseguidos por los subconjuntos `A` y `C` fueron del 69,31% y 69,48% mientras que los subconjuntos `B` y `D` únicamente obtuvieron unos porcentajes del 66,63% y ,.37% de acierto.

REPTree

Los resultados conseguidos con el árbol de decisión `RepTree` mejoraron los conseguidos por el otro árbol de decisión (`C4.5`) y también los conseguidos por el mismo algoritmo en el Experimento 5. Los resultados fueron los siguientes: el mejor resultado lo obtuvo el subconjunto `A` con un acierto del 69,43%, seguido por el subconjunto `C` con el 68,89% de acierto, mientras que los subconjuntos `B` y `D` únicamente consiguieron unos porcentajes del 66,03% y 65,34% de acierto.

ConjuctiveRule

Los resultados obtenidos con este algoritmo fueron especialmente relevantes en los subconjuntos sin cuotas pues mejoraron mucho con respecto a los mismos subconjuntos en el Experimento 5, en concreto el resultado de los subconjuntos `B` y `D` fue del 66.11% y 66.06% de acierto mientras que en el Experimento 5 apenas se llegó al 60.21% de acierto en este caso conseguido por el subconjunto `D`.

Los resultados obtenidos por los subconjuntos `A` y `C` fueron los segundos mejores por detrás de los obtenidos con el algoritmo `DecisionTable` con un porcentaje de acierto del 70.20% y 70.19% respectivamente.

6.3.6.4 Resultados del experimento

A continuación, se muestran algunos gráficos que servirán para comparar los resultados obtenidos con los distintos subconjuntos de datos y algoritmos aplicados. Además, también se podrán comparar los resultados de este experimento con los del Experimento 5 para de esta forma poder observar si el enfrentar los atributos fue positivo.

El gráfico de la Tabla 28 es válido para comprobar en líneas generales que subconjunto se comportó mejor, y además, ver si se mejoraron los resultados con respecto al Experimento 5. En este caso, el subconjunto que a nivel medio obtuvo mejores resultados fue el subconjunto `C` seguido muy de cerca por el subconjunto `A`, es decir, los dos subconjuntos que mejores resultados obtuvieron fueron los que en su conjunto de datos contaban con atributos con información de las cuotas de los distintos partidos. Además, se observa cómo en este experimento se consiguieron mejorar claramente los resultados de los subconjuntos en los que no había atributos con datos de cuotas. También se deduce que los resultados de los subconjuntos `A` y `C` no fueron mejorados debido a que la mayoría de los modelos de clasificación

generados tenían como principales atributos para clasificar las cuotas y los atributos de `CabezaDeSerie` los cuales son iguales en los experimentos 5 y 6.

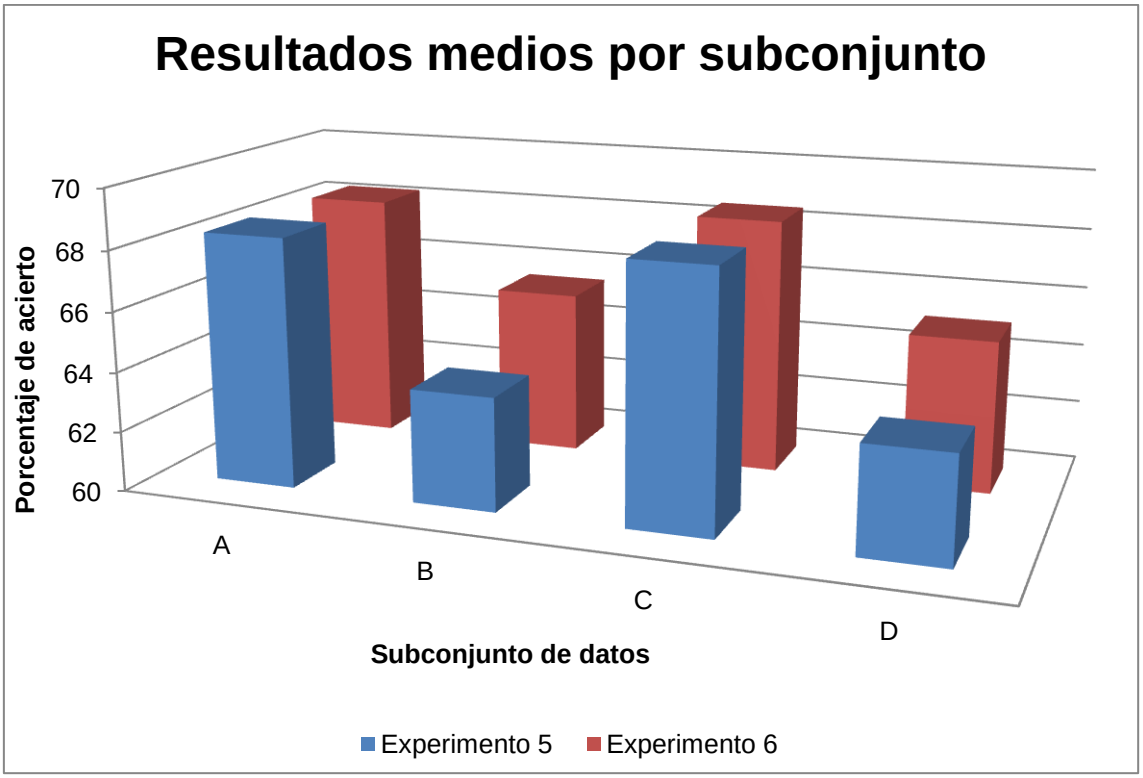


Tabla 28: Comparación de resultados por subconjuntos del experimento 6.

En el gráfico de la Tabla 29 se pueden comparar de forma visual todos los resultados, de esta forma se aprecia que, a excepción del algoritmo `C4.5`, los resultados iban a la par en los subconjuntos `A` y `C`, es decir, los subconjuntos con datos de cuotas, y los subconjuntos `B` y `D`, en los que no hay datos de cuotas. Además, se aprecia una ligera mejoría de los subconjuntos `A` y `B` con respecto a los subconjuntos `C` y `D` es decir, que se obtuvieron unos resultados ligeramente mejores en los subconjuntos en los que no se había realizado selección de atributos, también la clara excepción a esto se presentó con el algoritmo `C4.5` donde los subconjuntos `A` y `B` fueron superados por `C` y `D` respectivamente.

Por último, se muestra el gráfico de la Tabla 30 con la comparación de los resultados medios de los experimentos 5 y 6 para cada uno de los algoritmos, gracias a este gráfico se puede observar cómo, a excepción del algoritmo `AdaBoostM1` que fue el que peor resultados medios presentó, por norma general con la creación del nuevo subconjunto de datos con las estadísticas enfrentadas se mejoraron los resultados de forma notable, llegando a obtener mejoras de casi el 3% como con el algoritmo `ConjuctiveRule`.

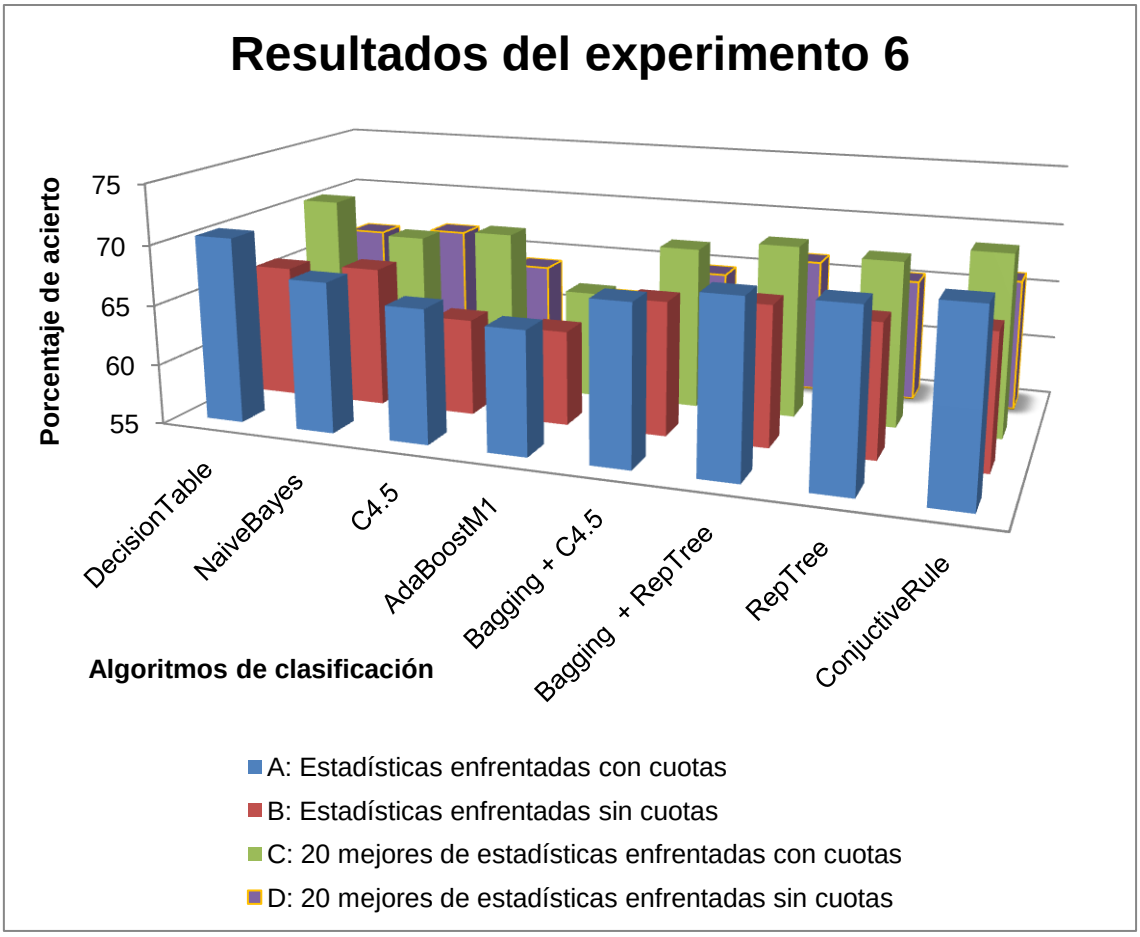


Tabla 29: Resumen de resultados del experimento 6.

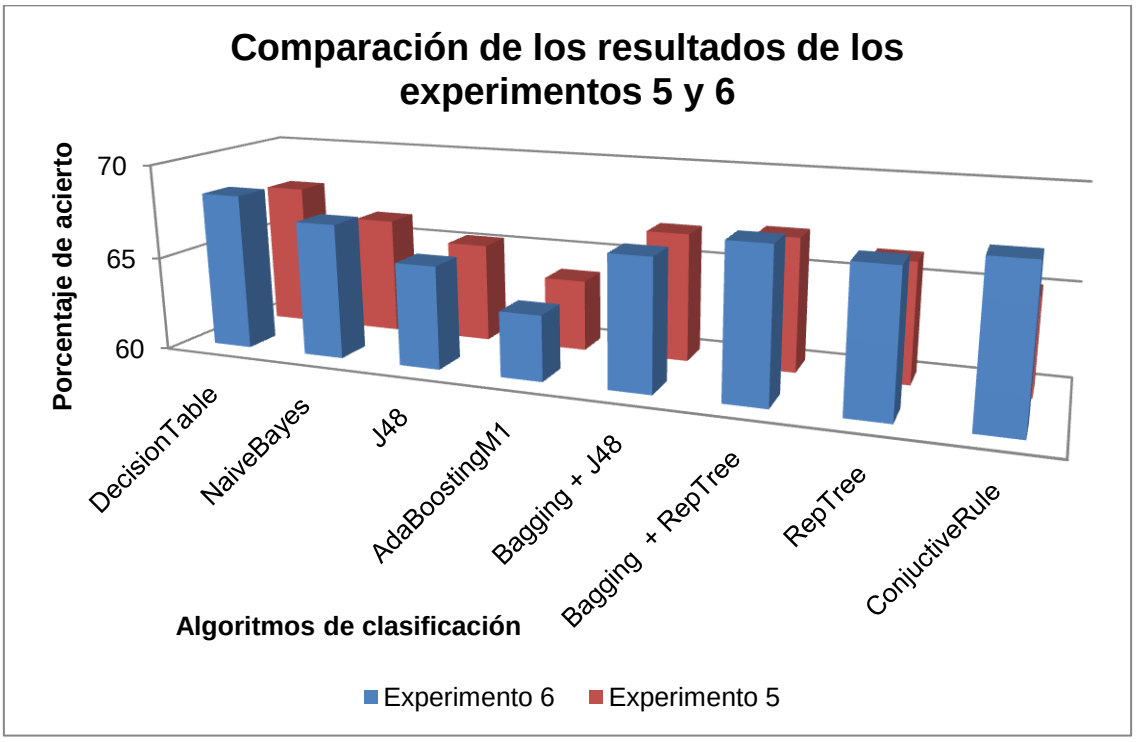


Tabla 30: Comparación de los resultados de los experimentos 5 y 6.

6.3.7 Experimento 7: Simulación real

Este experimento fue el último llevado a cabo en el proyecto. En él sólo se utilizó un algoritmo de clasificación, que fue el `DecisionTable`, esta decisión estuvo basada en la experiencia acumulada con el desarrollo de los experimentos anteriores en los cuales este algoritmo consiguió los mejores resultados. No se experimentó con más algoritmos de clasificación. Sin embargo sí que se realizó una primera prueba probando distintos métodos de búsqueda para el algoritmo `DecisionTable`. Una vez realizada esta prueba fue seleccionado el método de búsqueda con el cual se obtenían mejores resultados, y éste fue el utilizado en los siguientes pasos del experimento.

6.3.7.1 Conjunto de datos

En este experimento se cambió totalmente la forma de realizar la experimentación. Como punto de partida se dejó atrás la validación cruzada y se realizaron los test con un subconjunto de datos distinto al conjunto de entrenamiento.

Se tenían datos de apuestas desde junio de 2004 hasta marzo de 2009, por tanto se tomó la decisión de hacer una simulación real desde comienzos de 2008 y hasta marzo de 2009. Para realizar esta simulación se crearon 14 conjuntos de datos de entrenamiento con sus correspondientes conjuntos de test. En estos conjuntos se almacenaban los datos de todos los partidos disputados hasta el comienzo de un mes y en su correspondiente conjunto de test se encontraban los partidos del mes en cuestión. Por ejemplo, en el primero de los conjuntos de entrenamiento se encontraban todos los partidos disputados hasta enero de 2008 (no inclusive) y en su correspondiente fichero de test se encontraban todos los partidos disputados durante el propio mes de enero del 2008. De esta forma se formaron los 14 pares de ficheros correspondientes a 11 meses de 2008 (en diciembre no se disputaron partidos) más los tres meses de 2009.

Por los resultados obtenidos en experimentos anteriores se tomó la decisión de no aplicar selección de atributos. Además, se tuvieron en cuenta los conjuntos de datos con las estadísticas enfrentadas pues en los casos en los que se experimentaba sin los datos de las cuotas los resultados podrían ser mejores a los obtenidos con los atributos separados. Así pues, se realizó a la par el experimento para cuatro subconjuntos de datos cada uno de los cuales con sus 14 ficheros de entrenamiento y 14 ficheros de test. Los subconjuntos son los descritos a continuación:

A. **Estadísticas completas con cuotas**: Este subconjunto de datos estaba formado por todos los atributos incluidos en la Tabla 90 (ver *Anexo E: Atributos del Experimento 5*).

B. **Estadísticas completas sin cuotas**: Formado por el conjunto anterior sin los ocho atributos con información pertenecientes a las cuotas.

C. **Estadísticas enfrentadas completas con cuotas**: Formado por todos los atributos incluidos en la Tabla 91 (ver *Anexo F: Atributos del Experimento 6*).

D. **Estadísticas enfrentadas completas sin cuotas**: Formado por el conjunto anterior sin los ocho atributos con información pertenecientes a las cuotas.

6.3.7.2 Algoritmos de clasificación

Como fue indicado anteriormente en este caso sólo se utilizó para la experimentación el algoritmo `DecisionTable`. Se utilizó el subconjunto de datos `C` para experimentar con distintos métodos de búsqueda ofrecidos por el algoritmo y con los resultados obtenidos se escogió el método de búsqueda para el resto de subconjuntos.

De esta forma, los métodos de búsqueda aplicados al subconjunto `C` fueron: `BestFist`, `GeneticSearch`, `GreedyStepwise`, `LinearForwardSelection`, `RankSearch` y `SubSetSizeForwardSelection`. Los resultados obtenidos en cada uno de los 14 meses en los que se realizaron las pruebas se detallan en el gráfico de la Tabla 31.

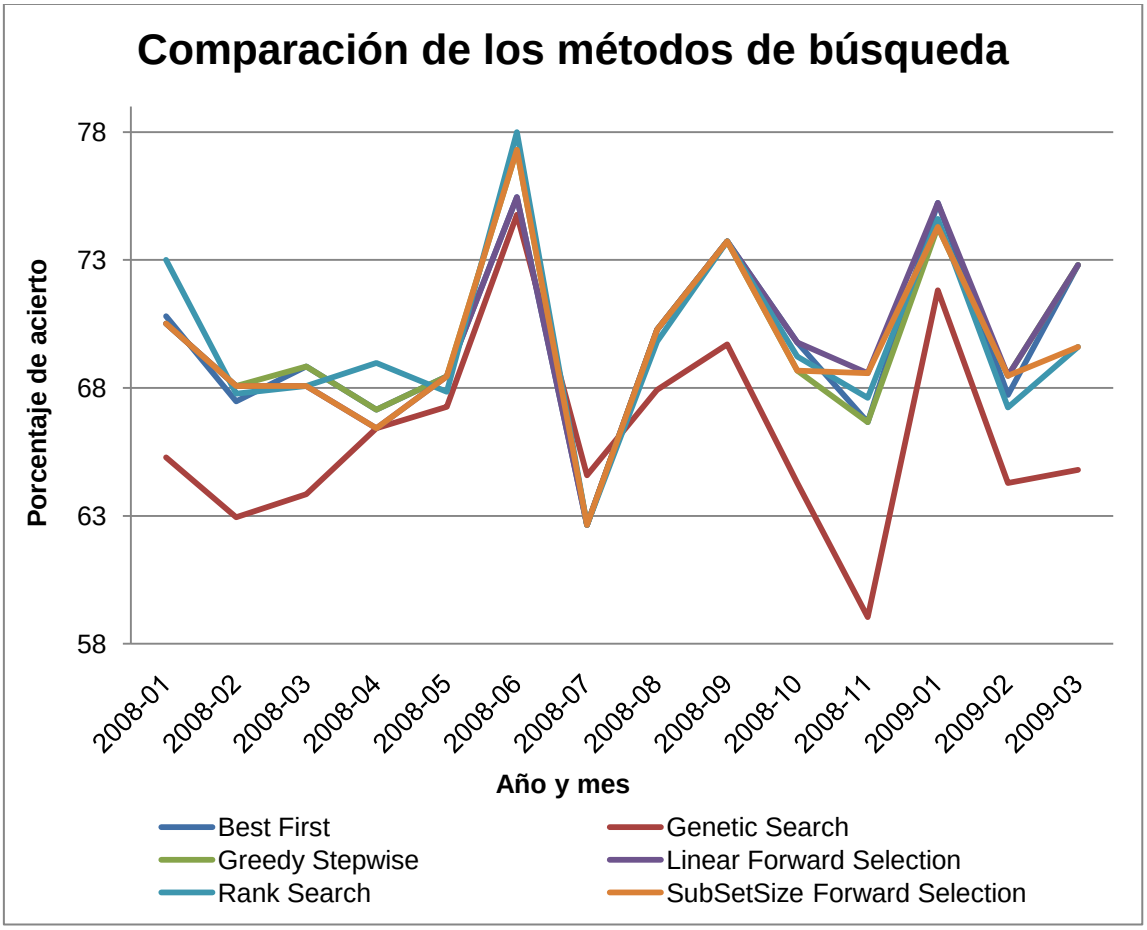


Tabla 31: Comparación de los métodos de búsqueda con DecisionTable para el subconjunto C.

Es difícil de apreciar por medio del gráfico anterior cuál de los métodos de búsqueda, como norma general se comportó mejor, para ello se muestra el gráfico de la Tabla 32 en el que se puede observar el porcentaje medio de acierto de cada uno de los métodos de búsqueda.

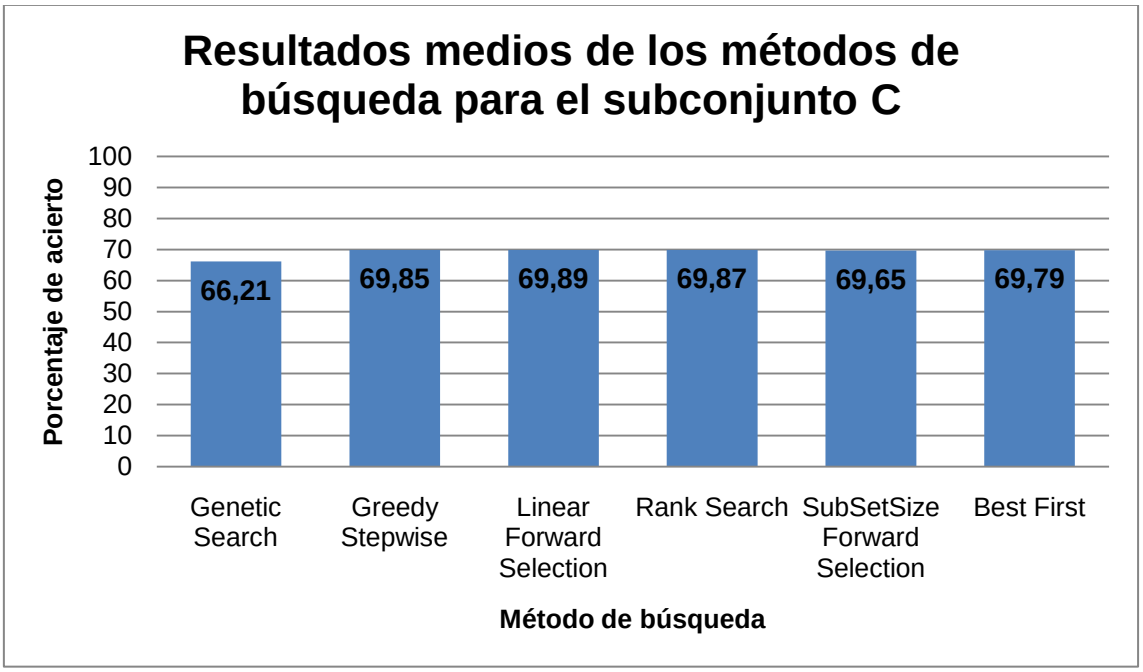


Tabla 32: Resultados medios de los métodos de búsqueda para el subconjunto C.

Como se puede apreciar los resultados fueron similares para la mayoría de los algoritmos siendo el que obtuvo mejores resultados el método de búsqueda `LinearForwardSelection`, que a partir de este momento fue el utilizado para el resto de subconjuntos.

A continuación, se ejecutó el algoritmo `DecisionTable` con el método de búsqueda seleccionado para el resto de los subconjuntos `A`, `B` y `D` siendo los resultados medios obtenidos del 69,74%, 64,23% y 66,80% de acierto respectivamente.

6.3.7.3 Resultados del experimento

En este apartado se presentarán gráficos en los que se pueden comparar los distintos resultados obtenidos para cada uno de los subconjuntos de datos. En ellos se puede ver claramente como la idea del Experimento 6 dio muy buenos resultados pues el subconjunto `D` mejoró claramente al subconjunto `B` en porcentaje de acierto.

Sin embargo, en los subconjuntos con cuotas apenas se apreciaron diferencias debido a la inclusión de los atributos de las cuotas en las reglas generadas por los algoritmos.

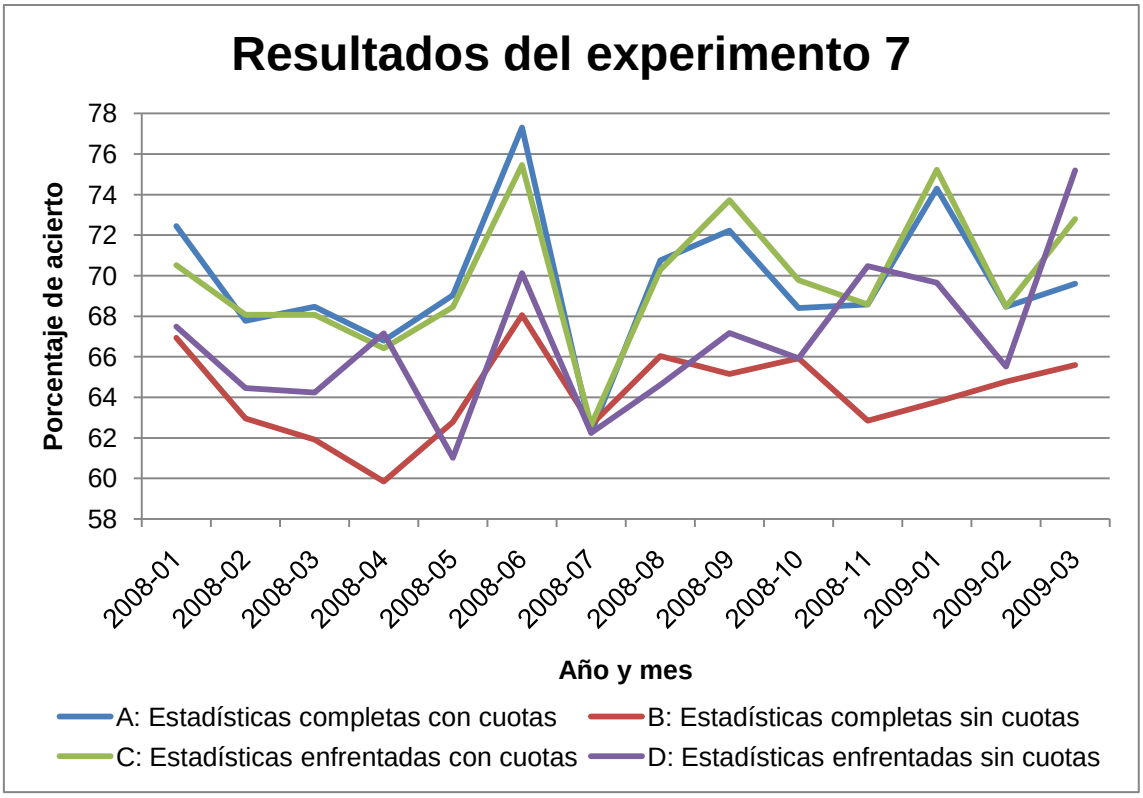


Tabla 33: Resumen de resultados del experimento 7.

Hay que destacar que en 3 de los 14 meses el subconjunto `D` es el que mejores resultados obtuvo, ya que este subconjunto carece de información de las cuotas, que siempre había sido determinante para elevar los porcentajes de acierto. Por último, se muestra el gráfico de la Tabla 34 con los resultados medios de cada uno de los subconjuntos en el que se pueden apreciar las diferencias entre los subconjuntos con cuotas y sin ellas y entre los subconjuntos con estadísticas completas y enfrentadas.

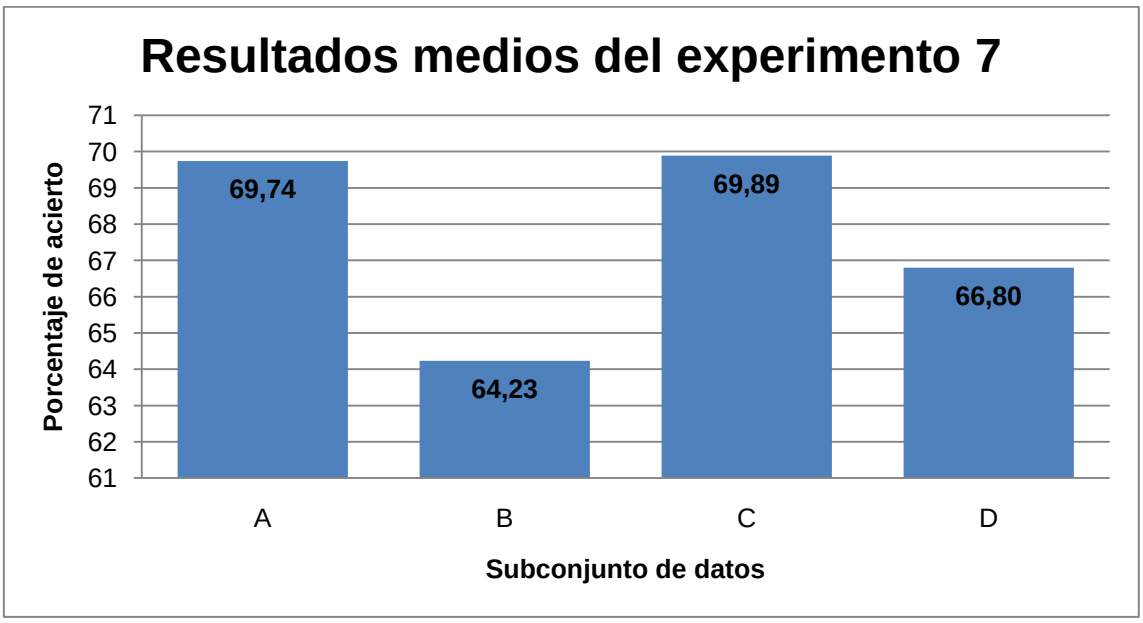


Tabla 34: Resultados medios del experimento 7.

6.4 Evaluación de los modelos

Antes del inicio de la experimentación se esperaban obtener mejores resultados de los conseguidos, alcanzando por lo menos el 80% de acierto. Sin embargo hasta la fase de evaluación, descrita en el próximo capítulo, no se pudo afirmar si los resultados eran buenos o malos, ya que para ello se tenían que hacer simulaciones de posibles inversiones y ver si se obtenían resultados positivos. De ser así, aunque, los porcentajes de acierto de los partidos no fueron todo lo bueno que se esperaba, unos resultados de ganancia positiva cumplirían uno de los objetivos planteados en el proyecto.

En este apartado se realiza un resumen de los resultados obtenidos en todos los experimentos realizados comparando unos con otros. El gráfico de la Tabla 35, indica los mejores resultados en cada uno de los experimentos en los que se utilizaron datos estadísticos sin cuotas.

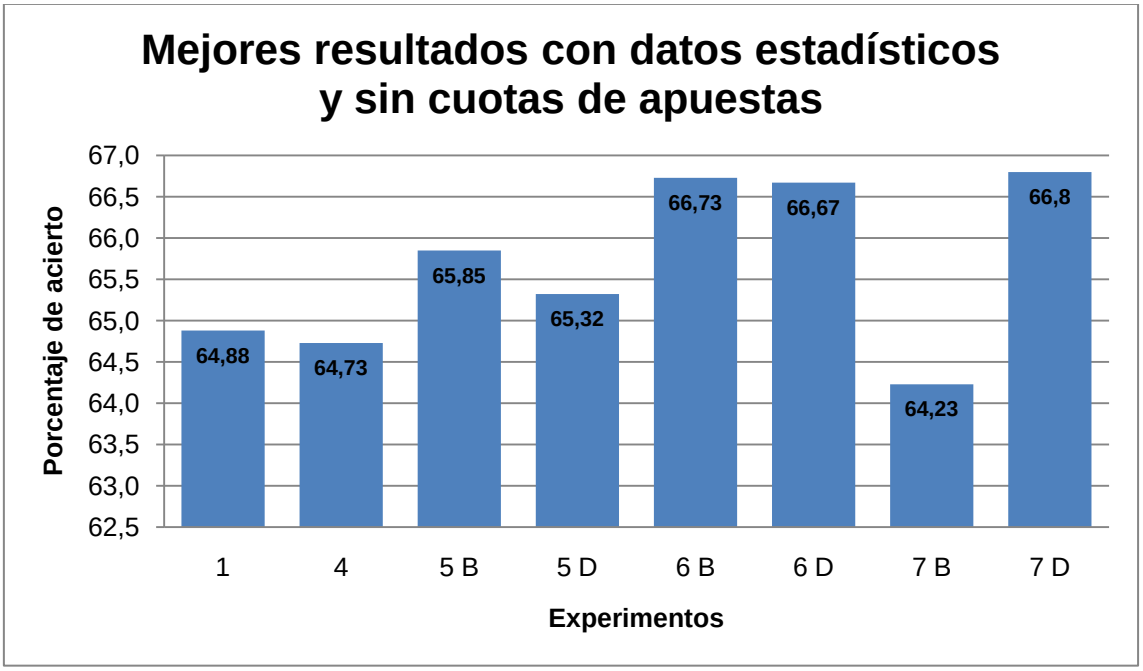


Tabla 35: Mejores resultados en los experimentos sólo con datos estadísticos de los jugadores.

A continuación, se muestra el gráfico de la Tabla 36, que es similar al anterior pero en este caso se reflejan los mejores resultados obtenidos por todos los experimentos en los que se utilizaron datos estadísticos, y además, datos con las cuotas de las apuestas. En los experimentos 5 y 6 los resultados fueron iguales, pues, los modelos generados eran también similares ya que estaban todos basados en las cuotas, las cuales no cambiaban de unos a otros.

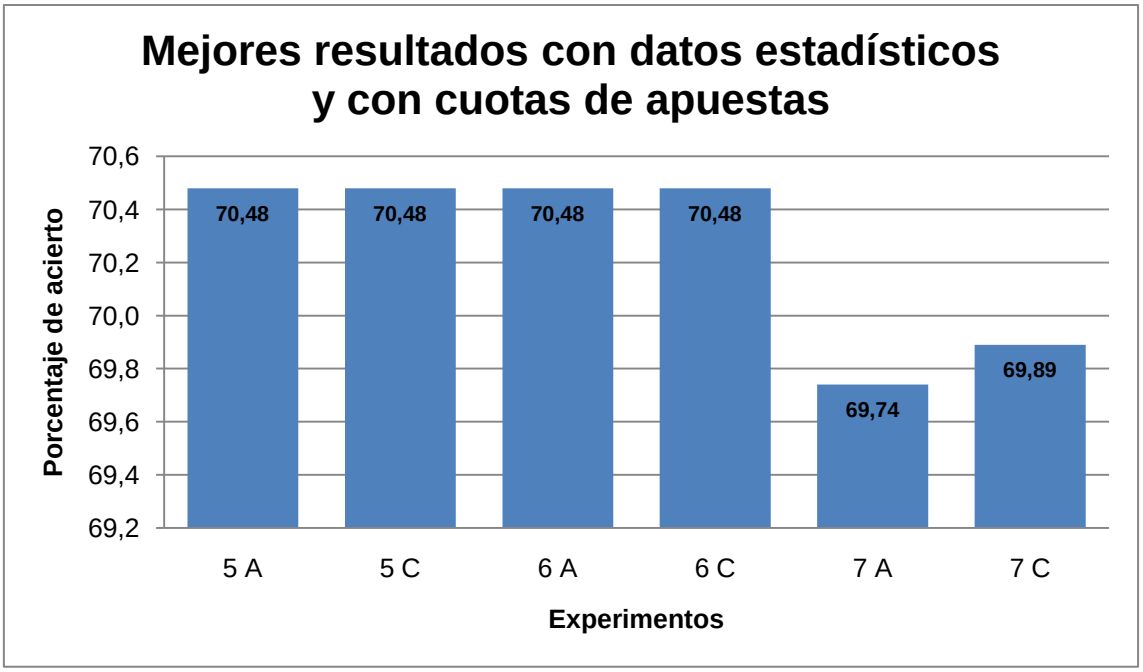


Tabla 36: Mejores resultados de los experimentos con datos estadísticos y de cuotas.

Por último, en el gráfico de la Tabla 37 se realiza una comparación entre el Experimento 7, que no fue realizado con validación cruzada, y los experimentos 5 A, 5 B, 6 A y 6 B. En este caso se puede observar como los resultados empeoraron en tres de los cuatro casos, sin embargo las diferencias no fueron significativas por ser mínimas en todos los casos.

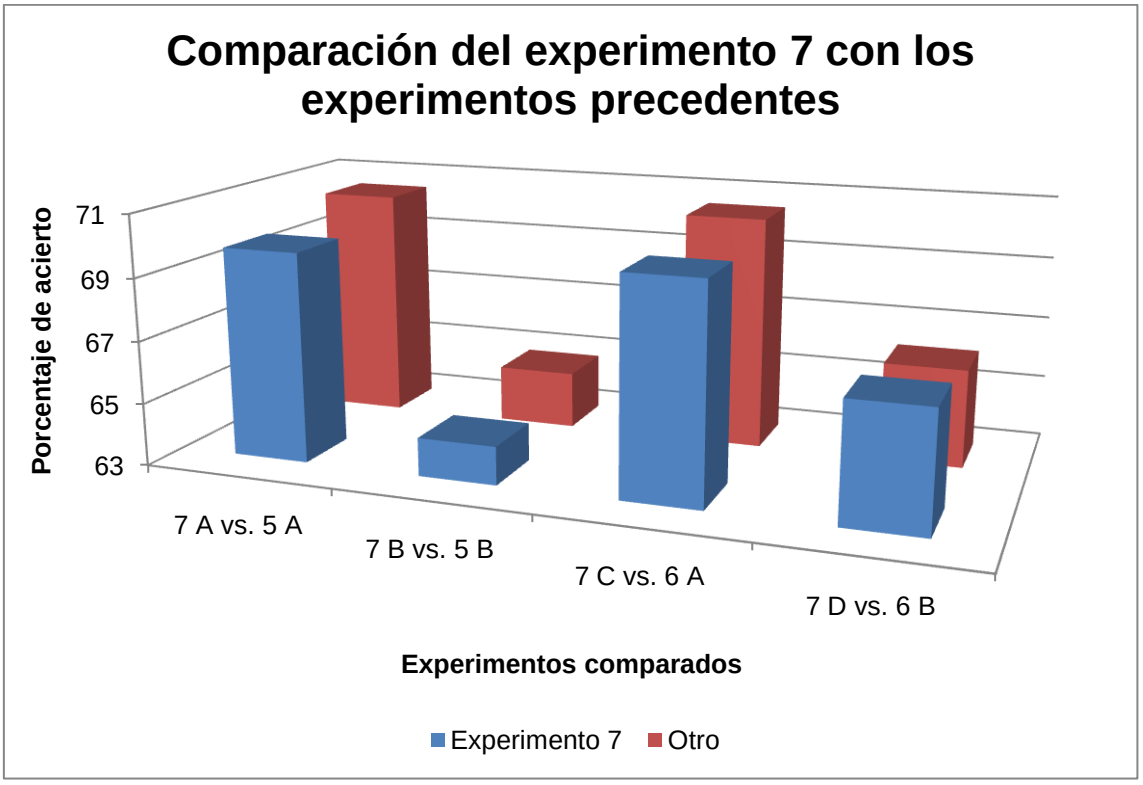


Tabla 37: Comparación del experimento 7 con los experimentos precedentes.

Capítulo 7: EVALUACIÓN

En este capítulo serán evaluados los modelos creados durante la fase de modelado. Se describirá la simulación de un proceso de inversión basado en los modelos, de forma que se pueda analizar si se obtienen resultados rentables económicamente. Se realizaron simulaciones utilizando distintos sistemas de apuestas. Todos los resultados y tablas recogidos en este capítulo se pueden encontrar en los archivos adjuntos en el CD del proyecto.

7.1 Sistemas de apuestas

Para realizar la evaluación de los modelos generados se podrían utilizar diversos sistemas, desde sistemas muy simples en los que se actúa siempre de la misma forma sin contemplar apenas variables, hasta sistemas más complejos en los que entran en juego la banca disponible, la probabilidad de éxito de determinado jugador en un partido, el número de apuestas que lleves perdiendo o ganando, etc.

Algunos de los sistemas planteados para la evaluación del proyecto se describen a continuación:

7.1.1 Apuesta fija

Es el método más simple de los utilizados. Únicamente utiliza para decidir si se lleva a cabo una apuesta la predicción realizada por el modelo generado. Siempre se apuesta una unidad a favor del jugador que el modelo clasifique como vencedor del partido sin tener en cuenta ni las cuotas, ni las probabilidades de éxito de las predicciones.

A priori no se esperaban tener grandes resultados con este sistema por lo que fue utilizado como toma de contacto para estimar la rentabilidad del modelo generado.

7.1.2 Basado en la cuota justa

Otro de los sistemas utilizados para evaluar los modelos generados está basado en el cálculo de la cuota justa a favor de cada uno de los jugadores. Como se ha descrito en apartados anteriores la cuota es inversamente proporcional a la probabilidad del resultado. Basándose en esta definición y en la probabilidad de que un jugador resulte ganador, obtenida del modelo generado, se puede calcular la que se considera cuota justa, es decir, a lo que tiene que estar pagada la victoria de ese jugador, por ejemplo:

Si en el modelo de predicción generado se predice como ganador al `jugador 1` con las siguientes probabilidades de victoria:

```
Jugador 1: 0,692

Jugador 2: 0,308
```

En este caso la cuota justa para el evento según el modelo sería:

```
Cuota Justa a favor del jugador 1 = 1 / 0.692 = 1,445
```

Una vez calculada la cuota justa se decide apostar o no a un evento si la cuota de mercado es superior a esta cuota. La regla más simple es apostar si la cuota encontrada es superior a la cuota justa, sin embargo, también se puede añadir un multiplicador que varíe el valor de la cuota a partir del cual apostar, aumentándolo más o menos. Por ejemplo, se apuesta a favor de un jugador si la cuota de mercado es superior a `(cuota justa * 1.05)`.

7.1.3 El criterio de Kelly

La fórmula del *"Criterio de Kelly"* fue desarrollada por John Kelly, un matemático aficionado a apostar a los caballos en 1956. Originalmente la formula está pensada para apostar a los caballos, pero es válida para cualquier tipo de apuesta. El sistema proporciona una fórmula que te ayuda a determinar el porcentaje de tu *bankroll* que debes destinar a una apuesta determinada. La fórmula es la siguiente:

```
      Porcentaje Bankroll = [(( Cuota x (probabilidad/100) ) - 1) / (Cuota - 1)] x 100

      Donde:
      Bankroll = Cantidad de dinero total que dispones para apostar
      Cuota = Cuota que ofrece la casa de apuesta
      Probabilidad = La estimación de la probabilidad de victoria del equipo o jugador
al que se va a apostar expresada en tanto por ciento (%).
```

Este sistema puede tener los siguientes problemas:

- La estimación de la probabilidad es algo subjetivo, por lo que la eficacia de este modo reside en la capacidad de asignar una probabilidad correcta al evento (lo que en el sistema anterior se llamó cuota justa). Se debe afinar más que los *oddsmaker*, que son las personas encargadas de determinar las cuotas de los eventos en las casas de apuestas tradicionales. Sin embargo, en este caso al trabajar con *Betfair* son los propios usuarios los que ponen las cuotas, por lo tanto hay que afinar más que el resto de usuarios.

- El porcentaje de *bankroll* a apostar puede llegar a ser un riesgo demasiado grande por lo que se aconseja usar multiplicadores que lo dividan.

7.1.4 Martingale

Este sistema no fue empleado en la evaluación del proyecto ya que todos los informes leídos acerca de su funcionamiento, ya sea en la ruleta de los casinos como en apuestas deportivas, eran negativos.

El Martingale es un sistema muy antiguo. Lo inventaron los jugadores de casino y es muy sencillo. Consiste en aumentar la apuesta cada vez que se pierde para compensar las pérdidas y obtener un beneficio al ganar la primera apuesta. Después de ganar la apuesta se vuelve a empezar desde el principio, apostando la cantidad inicial. Con la siguiente fórmula se puede calcular el valor de las apuestas:

```
Apuesta = (P+B) / (C-1)

Siendo:
P = Valor de las pérdidas anteriores acumuladas
B = Beneficio que se desea obtener
C = Cuota decimal
```

El problema surge ante una mala racha prolongada dado que el *stake* (`apuesta + dinero`) aumenta exponencialmente y no se tiene un *bankroll* ilimitado. Se puede pensar que en la ruleta es un mal sistema porque la probabilidad es del 50%, pero qué pasa si se usa este sistema con probabilidades de acierto mayor. Por ejemplo, en el tenis se apuesta a que Rafael Nadal o Roger Federer, los dos últimos números uno del ranking ATP, ganan. ¿Van a perder Rafael Nadal o Roger Federer 10 veces seguidas? Es de suponer que ni Rafael Nadal ni Roger Federer van a perder 10 partidos seguidos pero el problema en este caso es que la cuota en los partidos de Rafael Nadal y Roger Federer es muy improbable que sea superior a 2 en algún caso.

7.1.5 Apuesta proporcional

La apuesta proporcional (%) del total de la *bankroll* de apuestas no tiene los inconvenientes del sistema de apuesta fija. Apostando un porcentaje determinado del *bankroll* de apuestas se podrá:

- Variar el valor de las apuestas en función del tamaño del *bankroll*.
- Elegir el valor de las apuestas según las convicciones dentro de un margen.

Los expertos aconsejan apostar entre un 1% y un 5% del total del *bankroll* disponible.

7.2 Evaluaciones

Como es normal no tenía sentido evaluar todos los modelos generados, ya que si se generó en un experimento un modelo con un acierto del 70% y otro con el 60% es evidente que la evaluación del segundo de los modelos sería peor que la del primero.

Tampoco se pudieron evaluar todos los modelos de la misma forma, por ejemplo, en los modelos en los que se utilizaron los atributos de las cuotas para generarlos, sólo se podía utilizar en su evaluación la última cuota obtenida antes del inicio de los partidos, ya que, no se podía simular que se apostaba a la cuota máxima encontrada antes del inicio del partido porque en una simulación real el modelo se generaría en el último instante previo al inicio del partido, por lo que, la apuesta se realizaría en ese preciso instante y a la última cuota disponible antes del inicio.

Cosa distinta fue si para generar el modelo no se utilizaban las cuotas del partido, entonces se podían simular posibles inversiones apostando por ejemplo a la cuota máxima que se hubiera encontrado antes del inicio de los partidos.

7.2.1 Experimento 1

El primer experimento no pudo ser evaluado puesto que en él se utilizaron partidos que carecían de información relativa a las cuotas. En el apartado *6.3.1 Experimento 1: Sólo datos estadísticos* se describió que el conjunto de datos estaba formado por todos los partidos almacenados en la tabla `Partidos` de la base de datos independientemente de si estos estaban o no relacionados con alguna fila de la tabla de datos `Betfair`.

7.2.2 Experimento 2

Para realizar una primera toma de contacto con las evaluaciones de este experimento fueron evaluados dos modelos.

7.2.2.1 Modelo OneR

El modelo generado en este experimento consistía en una pequeña regla que determinaba uno u otro resultado según fuese la cuota mayor o menor a un valor determinado. Para ver si esta pequeña regla era interesante se generó un pequeño programa que la aplicase al fichero de test haciendo una simulación de apuestas fijas. El algoritmo del programa almacenaba la cuota y el resultado (ganada o perdida) de cada uno de los ejemplos de test y a continuación aplicaba la regla generada. En este caso, si la cuota era menor que 1.955 o estaba entre 1.9649999 y 1.975 se contabilizaba una apuesta aumentando un contador de apuestas en uno. A continuación, se examinaba el ejemplo para ver si había resultado ganadora y de ser así se sumaba la cuota a otro contador de ganancias. El resultado final fue:

```
Cantidad total apostada: 442179 €
Ganancia en las apuestas: 438746.37 €
Resultado total: -3432.63 €
```

Por tanto, el resultado obtenido con esta regla generó pérdidas con el sistema de apuestas fijas.

A continuación, se evalúa el mismo modelo con el sistema de apuestas proporcionales, suponiendo que al comienzo de la evaluación se disponía de un *bankroll* de 1000 €. Se realizaron cinco evaluaciones paralelas apostando en cada una de ellas el 1%, 2%, 3%, 4% y 5% del *bankroll* respectivamente. Los resultados obtenidos en todos los casos fueron negativos quedando el saldo del *bankroll* a 0.

```
Bankroll al final de la simulación en cada uno de los cinco casos:

bankroll 1%: 1.640702780194037E-35 €
bankroll 2%: 1.6065915530521996E-35 €
bankroll 3%: 1.588835350180656E-35 €
bankroll 4%: 1.570634013745739E-35 €
bankroll 5%: 1.5520047446760323E-35 €
```

7.2.2.2 Modelo C4.5:

Al realizar un test similar al realizado en el apartado anterior apenas se obtuvieron mejores resultados todavía generandose pérdidas. El resultado obtenido con el sistema de apuestas fijas fue el siguiente:

```
Cantidad total apostada: 446869 €
Ganancia en las apuestas: 443455.67 €
Resultado total: -3413.33 €
```

También se evaluó el modelo utilizando el sistema de apuestas proporcionales con los mismos porcentajes, es decir, del 1% al 5% y los resultados volvieron a ser nulos pues se perdió toda la banca inicial.

```
Bankroll al final de la simulación en cada uno de los cinco casos:

bankroll 1%: 9.940969938933713E-36 €
bankroll 2%: 9.734291015919369E-36 €
bankroll 3%: 9.626706704424055E-36 €
bankroll 4%: 9.516425341746941E-36 €
bankroll 5%: 9.403551147808925E-36 €
```

7.2.3 Experimento 3

Este experimento fue muy similar al anterior aunque los resultados del porcentaje de acierto fueron ligeramente peores. Se evaluó el modelo generado por el algoritmo `DecisionTable` el cual consistía en una pequeña regla que indicaba que si la cuota del evento era menor de 1.965 la apuesta resultaría ganadora.

Los resultados de la evaluación con el sistema de apuesta simple fueron los siguientes:

```
Cantidad total apostada: 326721 €
Ganancia en las apuestas: 323816.82 €
Resultado total: -2904,18 €
```

Los resultados conseguidos fueron mejores respecto a los del Experimento 2, sin embargo, seguían generando pérdidas. También se procedió a la evaluación del modelo con el sistema de apuestas proporcionales como en el Experimento 2. Al inicio de la simulación se contaba con un *bankroll* de 1000 € que acabaron perdiéndose después de la simulación siendo los resultados los mostrados a continuación:

```
Bankroll al final de la simulación en cada uno de los cinco casos:

bankroll 1%: 7.205260698339295E-28 €
bankroll 2%: 7.47647142231432E-28 €
bankroll 3%: 7.6074940764008055E-28 €
bankroll 4%: 7.7352680969523675E-28 €
bankroll 5%: 7.85965749769242E-28 €
```

7.2.4 Experimento 4

Al igual que en el Experimento 1 no pudo ser evaluado puesto que en él se utilizaron partidos que carecían de información relativa a las cuotas por tanto este experimento no fue evaluado económicamente hablando.

7.2.5 Experimento 5

El Experimento 7 es un experimento más avanzado que el 5, en él se utilizaron los datos que obtuvieron mejores resultados en el Experimento 5, y además, se hizo una simulación más real pues no se juntaron los datos para luego hacer validación cruzada con todos ellos sino que se hizo una simulación mes a mes actualizando los modelos generados todos los meses. La evaluación de este experimento se realizó, por tanto, como parte del Experimento 7 y podrá ser consultada en el apartado *7.2.7 Experimento 7*.

7.2.6 Experimento 6

Al igual que se señaló en el apartado anterior y por los mismos motivos la evaluación de este experimento fue realizada con la del Experimento 7 en el apartado *7.2.7 Experimento 7*.

7.2.7 Experimento 7

En este experimento se realizó la evaluación más completa de todas. Se utilizaron tres sistemas de apuestas: sistema de apuestas fijas, sistema basado en la cuota justa y el criterio de Kelly. Se dividieron las evaluaciones en cuatro partes, correspondientes a los cuatro subconjuntos con los que se realizó el experimento. Para la evaluación de todos los subconjuntos se realizaron distintos programas en Java, en ellos se introducían los modelos generados para cada uno de los meses y se invertía en cada uno de los partidos del mes siguiente aplicando el modelo correspondiente y los distintos sistemas de apuestas. De esta forma, después de los 14 resultados obtenidos para los 14 meses en los que se realizaba la simulación se pudo ver si se había conseguido generar un método ganador.

7.2.7.1 Evaluación del subconjunto A: Estadísticas completas con cuotas

Para generar los modelos se utilizaban datos de las cuotas que en una simulación real serían tomados antes del inicio de los partidos, por tanto, sólo quedaba la posibilidad de apostar a la última cuota registrada, a la que se apostó antes del inicio del partido, que es lo que se hizo en las distintas simulaciones.

A continuación, se muestran varias tablas con los resultados de la evaluación. En la Tabla 38 se refleja el resultado de la evaluación con el sistema de apuestas fijas, en este caso se simuló que se apostaba una unidad en cada partido a favor del jugador que predecía el modelo generado en el experimento por el algoritmo `DecisionTable`. La cuota a la que se apostaba era la última registrada en la base de

datos a favor de ese jugador antes del comienzo del. Además, también se ofrece el resultado obtenido si se eliminasen los picos de los resultados parciales, es decir si se eliminasen los resultados de los meses en los que mejor y peor resultado se obtuvieron. La inversión representa el número de partidos a los que se apostó, que en este caso son todos los partidos de 2008 y 2009 y la predicción errónea indica el número de partidos en los que falló la predicción del modelo generado.

Resultados de la evaluación con el sistema de apuestas fijas	
Balance de inversiones a la última cuota	-16,26
Balance de inversiones eliminando el mejor y el peor mes	-22,16
Inversión (Número de partidos)	3.987
Predicción errónea (Número de partidos)	1.193

Tabla 38: Resultado de la evaluación del experimento 7 A con el sistema de apuestas fijas.

Como se puede observar el resultado final fue de pérdidas a lo largo de los 14 meses acabando con un balance negativo de -16.26 unidades monetarias. El resultado del balance eliminando los picos también fue negativo.

La siguiente evaluación se realizó con el sistema de apuestas basado en la cuota justa. Nuevamente se muestran los resultados totales de la simulación a lo largo de los 14 meses y los resultados obtenidos eliminando los picos. Además, se muestran 20 resultados distintos fruto de aplicar un multiplicador que aumentase el valor de la cuota a la que se apostaba. Por ejemplo, si la cuota justa calculada era de 1,5 y el multiplicador era 1,1 únicamente se apostaba al partido si la última cuota registrada a favor del jugador que el modelo indicaba como ganador superaba (`1,5 x 1.1 = 1,65`). Los resultados obtenidos se muestran en la Tabla 39:

Multiplicador	Balance	Balance sin picos	Rentabilidad	Invertido (Nº de Partidos)	Predic. errónea
1	47,99	33,21	1,52%	2.191	700
1,01	54,08	36,17	1,84%	1.967	635
1,02	40,46	28,01	1,57%	1.783	594
1,03	35,83	25,49	1,61%	1.582	541
1,04	31,33	18,24	1,34%	1.365	478
1,05	44,9	30,83	2,70%	1.141	405
1,06	39,29	21,53	2,26%	952	355
1,07	41,16	25,27	3,11%	812	309

Multiplicador	Balance	Balance sin picos	Rentabilidad	Invertido (Nº de Partidos)	Predic. errónea
1,08	39,82	31,24	4,46%	700	272
1,09	39,31	32,88	5,49%	599	238
1,1	33,85	26,56	5,23%	508	210
1,11	31,31	25,03	5,82%	430	183
1,12	32,52	28,56	7,56%	378	162
1,13	26,28	19,78	6,03%	328	147
1,14	24,11	18,81	6,25%	301	138
1,15	25,96	15,08	5,89%	256	120
1,16	33,73	20,06	9,46%	212	97
1,17	36,2	21,31	11,52%	185	84
1,18	25,38	12,31	7,84%	157	77
1,19	47,99	16,18	11,72%	138	67

Tabla 39: Resultado de la evaluación del experimento 7 A con el sistema de apuestas basado en la cuota justa.

A continuación, se muestra el gráfico de la Tabla 40 en donde se muestra más claramente la evolución del balance con los distintos multiplicadores. El mejor multiplicador fue el 1,01 y a medida que se aumentaba el valor del multiplicador parecía que los resultados empeoraban con ciertas alternancias de subidas y bajadas.

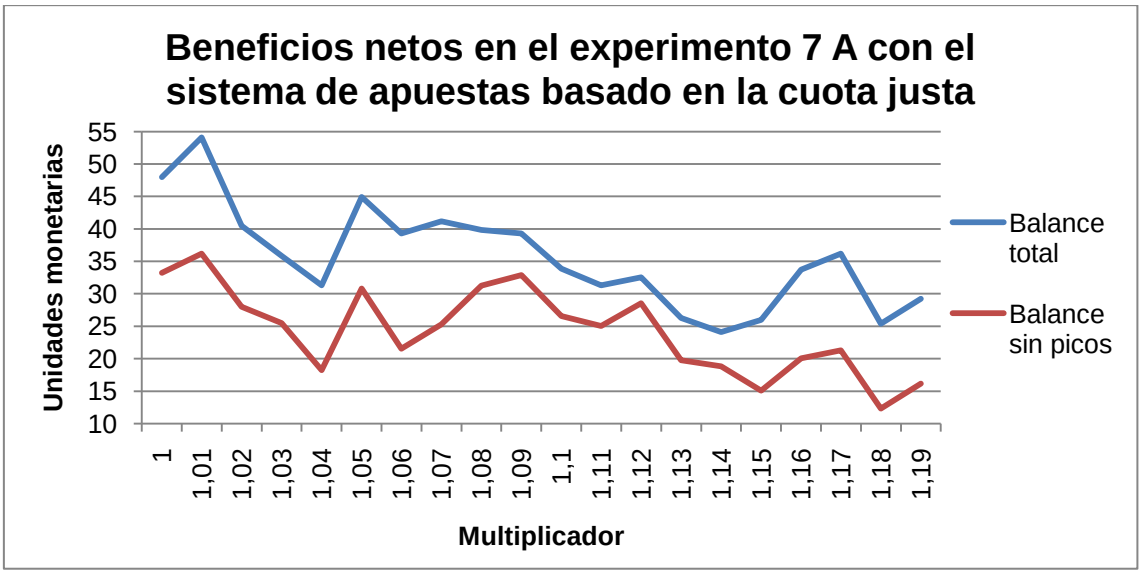


Tabla 40: Comparación de los balances con los distintos multiplicadores en el sistema de apuestas basado en la cuota justa del experimento 7 A.

Los resultados de la evaluación de este experimento fueron positivos obteniendo en el mejor de los casos un beneficio neto de 54,08 unidades monetarias. Sin embargo, éste fue el mejor resultado neto pero no el que mayor rentabilidad ofrecía. El término rentabilidad hace referencia a los beneficios obtenidos por partido apostado. A continuación, se muestra el gráfico de la Tabla 41, en el que se puede ver como la rentabilidad aumentaba a medida que el valor del multiplicador también aumentaba. Esto fue debido a que al aumentar el valor del multiplicador las condiciones para apostar a un evento se endurecían, ya que, suponía que la cuota a la que se apostaba tenía que ser mayor que con un multiplicador menor, por lo tanto el número de partidos al que se apostó fue menor a medida que el multiplicador aumentaba su valor.

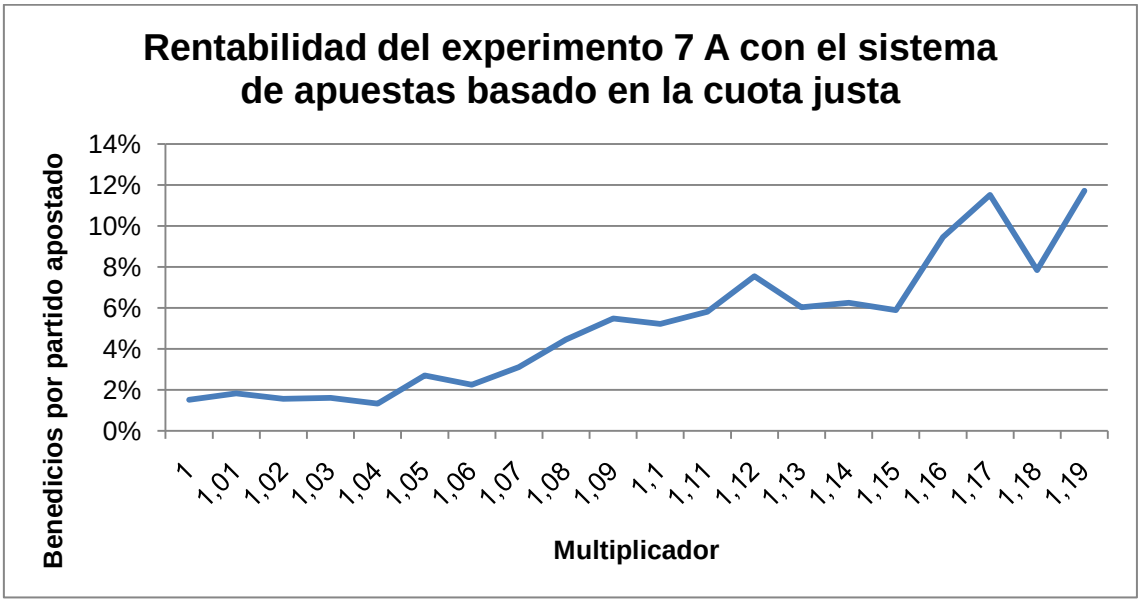


Tabla 41: Evolución de la rentabilidad en función del multiplicador aplicado en el experimento 7 A con el sistema de apuestas basado en la cuota justa.

Por último, se realizó la evaluación del experimento aplicando como sistema de apuestas el criterio de Kelly. Se estableció al comienzo de cada uno de los meses una banca de 1000 unidades monetarias. Al comienzo de cada mes la banca volvía a estar en 1000, ya sea porque se habían retirado beneficios o porque se había aumentado porque hubiese producido pérdidas en el mes anterior. Esto se hizo de esta forma porque si la banca aumentaba mucho se tenían que hacer apuestas con una gran inversión, que además de ser un riesgo por el propio dinero a invertir, podían producir la situación de que el inversor se quedase sin volumen de mercado, es decir, que no existiese el suficiente número de usuarios con el suficiente dinero que apostasen en contra de lo que el inversor apostaba. Además, en este caso también se utilizaron porcentajes para variar el *bankroll* utilizado en cada apuesta y al final comparar resultados y saber con qué multiplicador se obtenían mayores beneficios. Todos los resultados se pueden ver en el *Anexo G: Resultados de las evaluaciones con el criterio de Kelly*.

A continuación, se muestra el gráfico de la Tabla 42 con la evolución de los beneficios según el multiplicador aplicado:

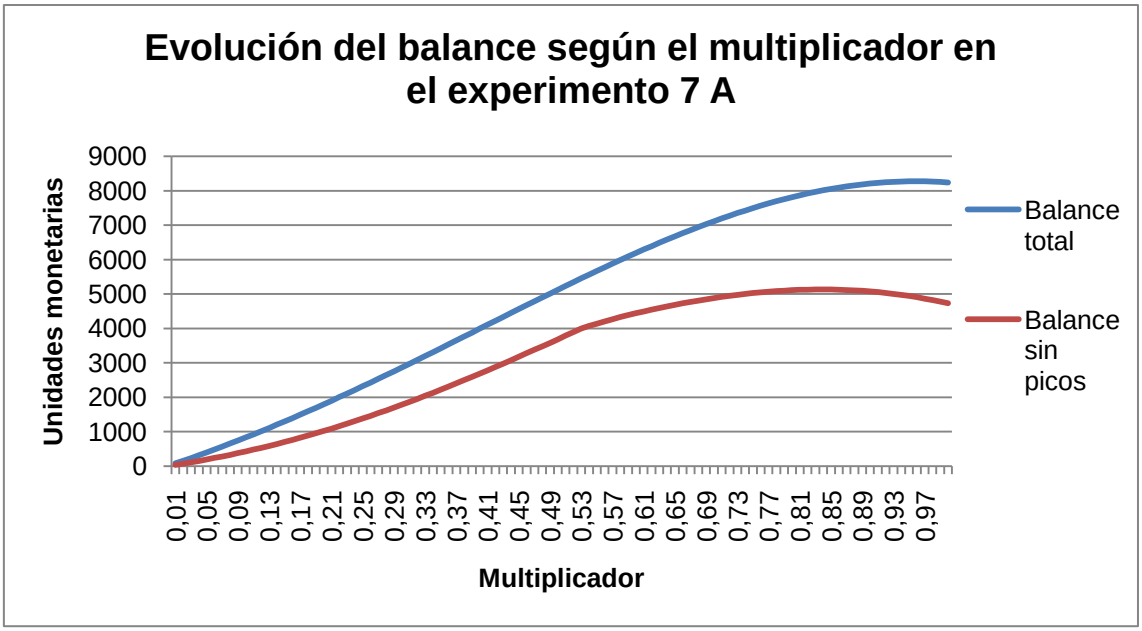


Tabla 42: Comparación del balance con los distintos multiplicadores utilizando como sistema de apuestas el criterio de Kelly en el experimento 7 A.

Como se puede observar los resultados obtenidos con este sistema fueron muy buenos llegando a obtener después de los 14 meses unos beneficios netos de 8.278,36 unidades monetarias en el mejor de los casos. Sin embargo, si se observan detenidamente los ficheros de resultados adjuntos en el CD del proyecto es curioso ya que se perdió dinero (la banca inicial de 1000 disminuyó al final de mes) en 9 de los 14 meses en los que se invirtió, sin embargo, en los 5 meses en los que se obtuvieron beneficios esas pérdidas fueron claramente compensadas. También es evidente observando el gráfico que los multiplicadores que mejores resultados ofrecieron fueron los cercanos a 1.

Llama la atención la diferencia que hubo entre el balance total y el balance eliminando los picos, la explicación a estos resultados fue evidente, al eliminar los picos no se tenían en cuenta los meses con mejor y peor resultado. En este caso el mes con peor resultado nunca podía tener pérdidas de más de 1000 unidades puesto que era la inversión inicial al principio de cada mes, sin embargo, el mejor de los meses si podía aumentar la banca en un porcentaje muy elevado. En este caso el mes con mejores resultados fue junio de 2008 en el que a final de mes se dejó la banca con 5.498,85 unidades monetarias después de haber apostado en 138 partidos en los cuales la predicción sólo falló en 38 de ellos.

7.2.7.2 Evaluación del subconjunto B: Estadísticas completas sin cuotas

En este apartado se evalúa el experimento 7 B, en él no se utilizaron datos de las cuotas a favor de cada uno de los jugadores, por tanto se podía apostar en cualquier momento antes del comienzo de los partidos y desde el momento en el que el partido era ofrecido por la casa de apuestas, esto fue útil pues quedaron reflejadas las diferencias existentes de las apuestas realizadas a las cuotas media, mínima,

máxima y última. Por tanto, la desventaja que se tenía al tener un modelo que clasifica con un porcentaje de acierto menor al de los experimentos en los que sí se incluyen las cuotas en el conjunto de datos, podía ser compensada con la posibilidad de apostar a una cuota mayor a la encontrada en el último instante.

El primer sistema de apuestas utilizado para evaluar el modelo fue el de apuestas fijas. Se simuló que se apostaba una unidad, en cada uno de los partidos, a favor del jugador que el modelo generado determinaba como ganador. Se ofrecen cuatro resultados relacionados cada uno con el valor de la cuota a la que se apostó, en este caso la cuota media a favor del jugador desde el momento en que la casa de apuestas ofrece el partido hasta antes de que comience, la cuota máxima y mínima encontradas a lo largo de este periodo y la última cuota a la que se registraron apuestas en el instante anterior al comienzo del partido. Los resultados obtenidos varían mucho entre ellos, principalmente entre haber apostado con cuotas mínimas o máximas. Sin embargo, los resultados extremos nunca son una buena medida y para poder sacar partido a estos resultados habría que realizar otros estudios en los que se tratase de encontrar en qué momentos se produce la cuota máxima.

Descripción	Balance	Balance sin picos
Apostando a la cuota media	-57,23	-40,14
Apostando a la cuota máxima	1.483,68	466,93
Apostando a la cuota mínima	-424,08	-357,69
Apostando a la última cuota	-38,91	-30,08
Inversión (Número de partidos)	3.987	3.450
Predicción errónea (Número de partidos)	1.418	1.236

Tabla 43: Resultado de la evaluación del experimento 7 B con el sistema de apuestas fijas.

La siguiente evaluación del modelo se realizó con el sistema de apuestas basado en la cuota justa. En este caso la Tabla 44 es más grande que la mostrada para el mismo sistema en el subconjunto A, de este mismo experimento, ya que, se simularon las apuestas para cuatro cuotas distintas lo que conlleva por cada una de ellas cuatro columnas (balance, balance sin picos invertido y predicción errónea).Por tanto, sólo se mostrará en este caso una tabla con los balances netos, pudiéndose encontrar todos los resultados en los archivos adjuntos en el CD del proyecto. A simple vista se pueden sacar conclusiones sorprendentes como que el resultado de apostar a la última cuota registrada fue mejor que el de apostar a la cuota media. Los resultados obtenidos si se consiguiese apostar a la cuota máxima volvieron a ser muy buenos por lo que empezó a quedar claro que una de las posibles ampliaciones al proyecto podría ser la realización de un estudio en el que se intentase predecir cuándo se producen los picos positivos de las cuotas.

Multiplicador	Balance cuota media	Balance cuota máxima	Balance cuota mínima	Balance última cuota
1	6,75	1.450,36	-146,2	30,72
1,01	2,12	1.440,62	-152,19	37,24
1,02	4,47	1.439,67	-145,46	36,12
1,03	6,81	1.433,22	-133,48	33,51
1,04	-1,22	1.432,45	-127,42	30,09
1,05	-0,81	1.432,62	-128,37	29,15
1,06	-2,54	1.428,9	-126,94	23,84
1,07	2,38	1.419,9	-122,38	29,1
1,08	3,07	1.422,47	-115,65	30,03
1,09	5,93	1.418,68	-117,35	30,34
1,1	-4,48	1.430,14	-114,57	31,04
1,11	-10,44	1.429,36	-108,99	28,61
1,12	-3,53	1.420,54	-94,07	19,42
1,13	-9,52	1.407,67	-94,29	20,16
1,14	-14,03	1.409,87	-87,92	11,83
1,15	-9,98	1.399,97	-95,38	20,28
1,16	-3,79	1.393,55	-92,76	24,09
1,17	1,79	1.388,77	-87,12	28,77
1,18	-2,44	1.385,92	-86,22	26,3
1,19	0,65	1.384,69	-85,35	23,51

Tabla 44: Resultados de la evaluación del experimento 7 B con el sistema de apuestas basado en la cuota justa.

A continuación, se adjuntan gráficos comparativos en los que se pueden comparar los resultados obtenidos con los distintos multiplicadores para cada una de las cuotas a las que se simuló la realización de apuestas.

En la Tabla 45 se adjunta el estudio realizado con la cuota media en el que llaman la atención los saltos producidos entre los multiplicadores encontrando el mejor resultado con el multiplicador tomando el valor 1. Hay que destacar que el balance sin picos ofreció mejores resultados que el balance total, esto fue debido a que en el mes

en que peores resultados se obtuvieron se pierde más de lo que se gana en el mes con mejores resultados.

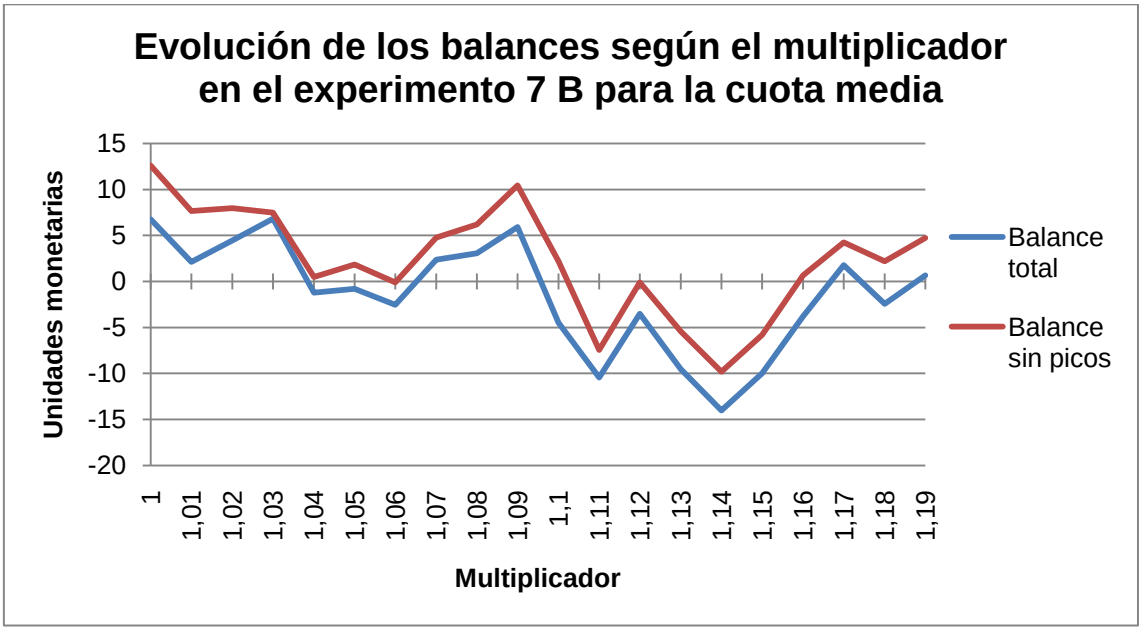


Tabla 45: Comparación del balance con los distintos multiplicadores con el sistema de apuestas basado en la cuota justa del experimento 7 B y con las apuestas realizadas a la cuota media.

Si se observa el gráfico de la Tabla 46 con los resultados obtenidos apostando a la cuota máxima, la ganancia obtenida en el mejor mes es muchísimo más grande que las pérdidas del peor mes, de hecho, los resultados totales se multiplicaban por más de tres al incluir el mejor mes, esto pudo ser debido a que hubiese algún usuario despistado que ofreciese una cuota equivocada que hubiese hecho aumentar los resultados del experimento de semejante forma.

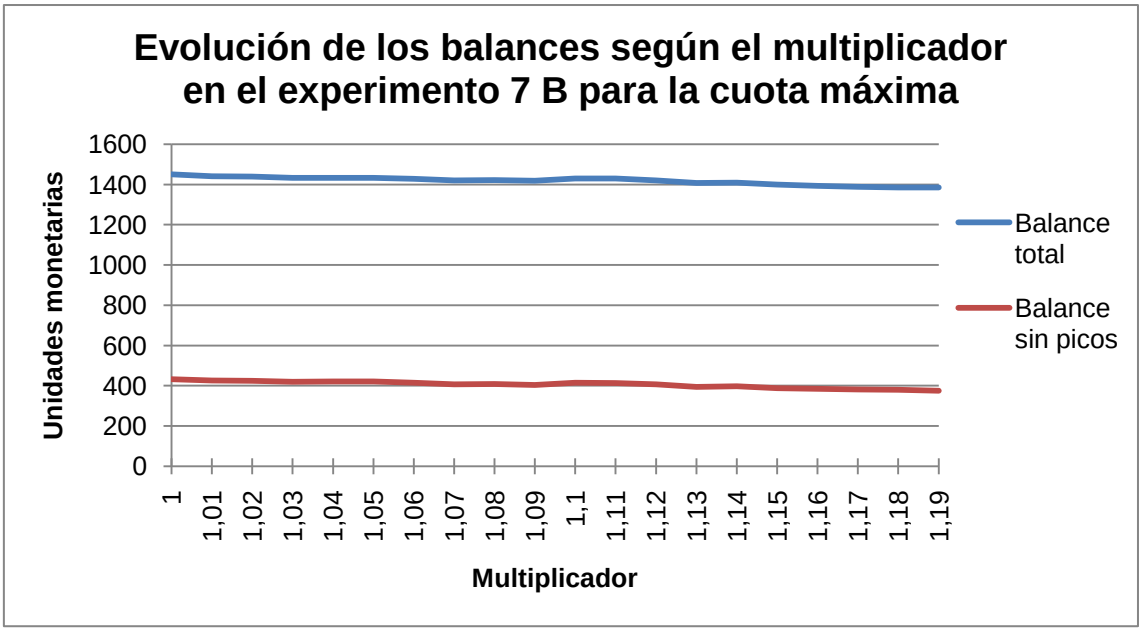


Tabla 46: Comparación del balance con los distintos multiplicadores con el sistema de apuestas basado en la cuota justa del experimento 7 B y con las apuestas realizadas a la cuota máxima.

Los resultados obtenidos apostando a la cuota mínima fueron en todos los casos negativos y sin grandes diferencias entre el balance total y el balance sin picos por lo que no se muestra la tabla con sus resultados. Algo similar ocurrió en el caso en que las apuestas eran realizadas a la última cuota registrada, al apostar a esta cuota se registraban mejores resultados que si se apostaba a la cuota media, y además, se contaba con la ventaja de que es el método más fácil de aplicar ya que no hay que realizar estudios de cuando se va a producir la cuota. Por tanto, el apostar a la última cuota puede ser una oportunidad de negocio pues los resultados fueron positivos y su aplicación es muy sencilla simplemente habría que aplicar el modelo generado para predecir el ganador y apostar a favor de la predicción en el último instante antes del comienzo del partido. A continuación, se muestra el gráfico de la Tabla 47 con los resultados obtenidos:

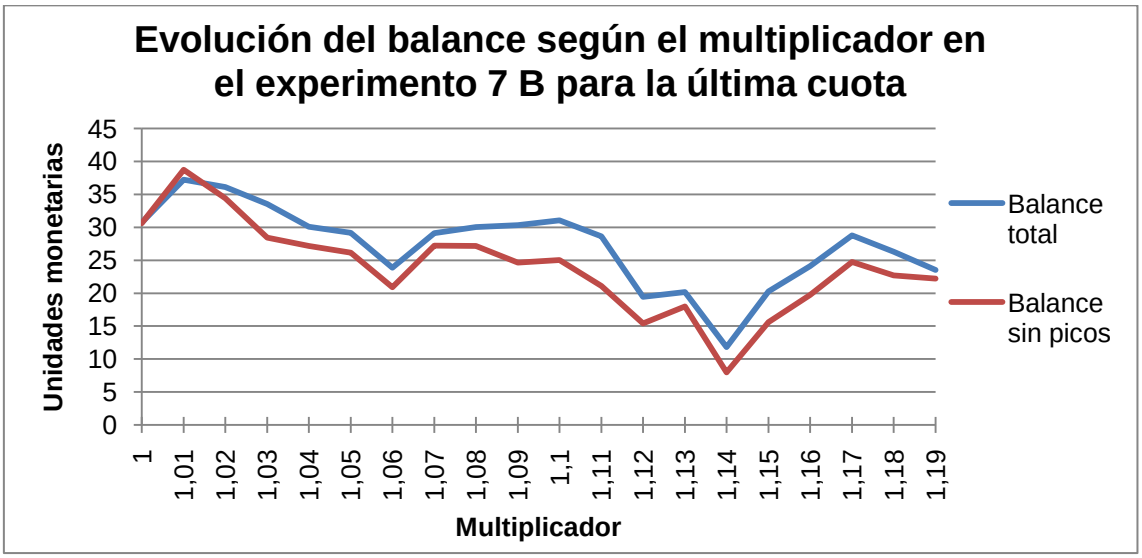


Tabla 47: Comparación del balance con los distintos multiplicadores con el sistema de apuestas basado en la cuota justa del experimento 7 B y con las apuestas realizadas a la última cuota.

Como ocurría con la cuota media los mejores multiplicadores fueron los más cercanos a uno, en este caso el multiplicador con el valor 1,01 fue el que obtuvo mejores resultados generando una ganancia en el balance total de 37,24 unidades monetarias.

A continuación, se presentan los resultados en términos de rentabilidad, es decir, el porcentaje de ganancia medio obtenido en cada partido apostado para cada una de las cuotas a las que se apuesta. El primero de los gráficos evidencia la rentabilidad de apostar a la cuota máxima y los pésimos resultados obtenidos en caso de apostar a la cuota mínima Sin embargo, no quedan claros los porcentajes de beneficio en caso de apostar a la última cuota o a la cuota media, por lo que se muestra un segundo gráfico sólo con estos resultados.

También cabe destacar en el gráfico de la Tabla 49Tabla 48 la mejora del porcentaje de beneficio al aumentar el valor del multiplicador. Sin embargo, si se observa el gráfico de la Tabla 46: Comparación del balance con los distintos multiplicadores con el sistema de apuestas basado en la cuota justa del experimento 7

B y con las apuestas realizadas a la cuota máxima. se puede ver que esto no está relacionado con el beneficio neto que disminuye al aumentar el multiplicador.

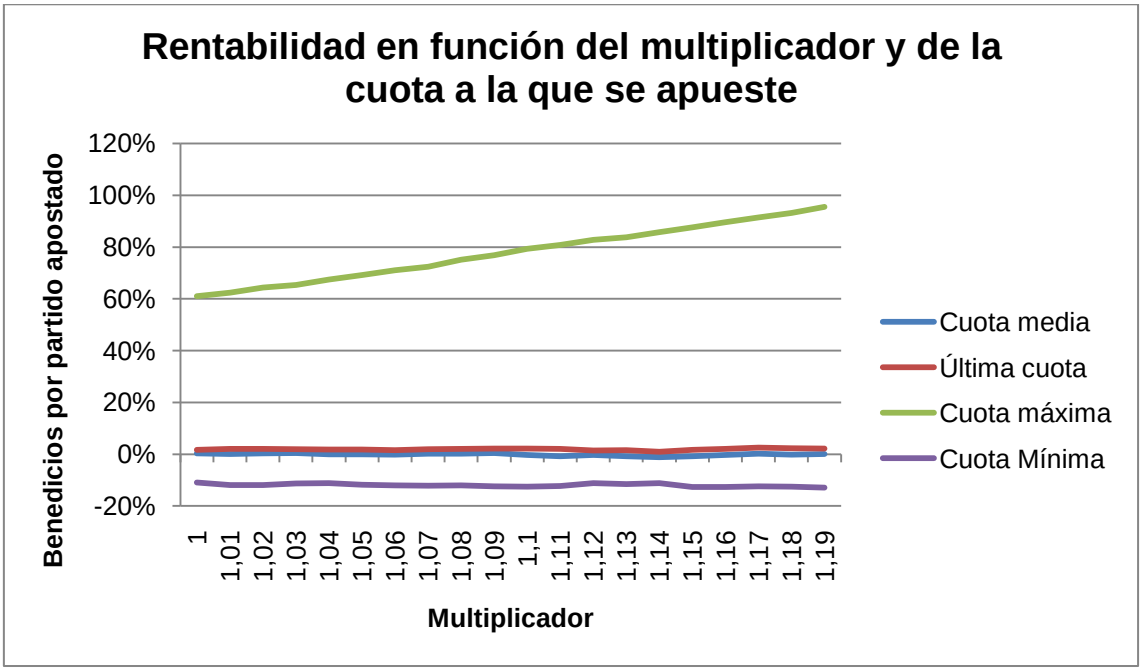


Tabla 48: Evolución de la rentabilidad en función del multiplicador aplicado en el experimento 7 B con el sistema de apuestas basado en la cuota justa.

En el gráfico de la Tabla 49 se puede observar como la rentabilidad conseguida al apostar a la última cuota fue positiva. Sin embargo, la conseguida si se apostase a la cuota media varíaba entre valores positivos muy bajos, nunca superando el 0.5%, y valores negativos.

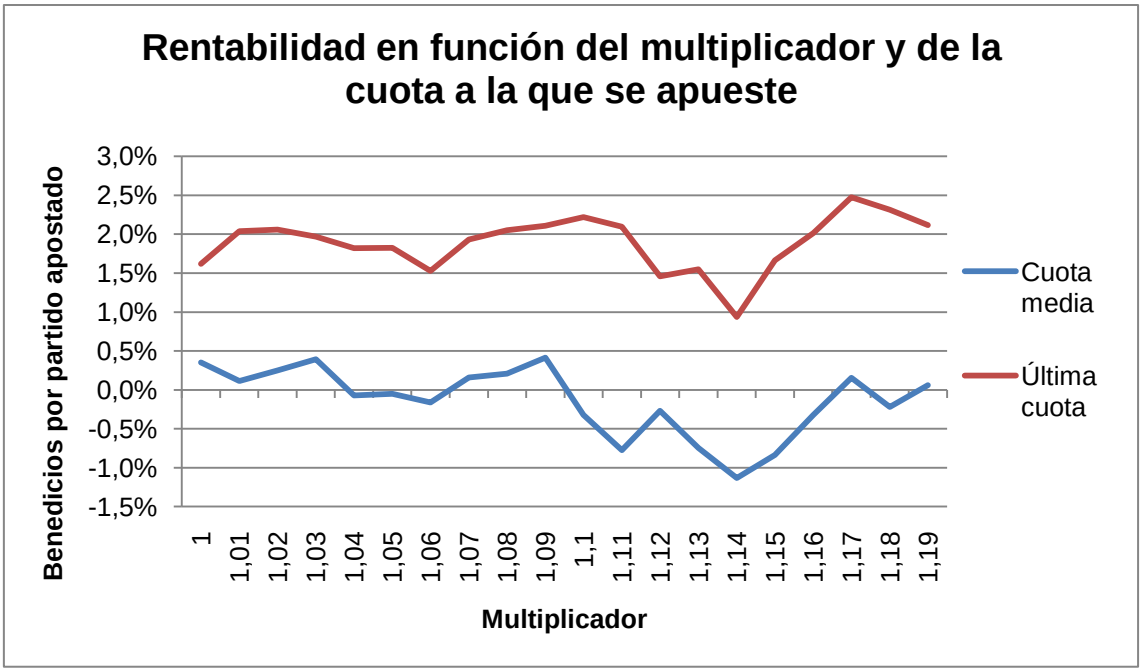


Tabla 49: Evolución de la rentabilidad en función del multiplicador aplicado en el experimento 7 B con el sistema de apuestas basado en la cuota justa para las apuestas realizadas a la última cuota y a la cuota media.

Por último para terminar la evaluación del experimento 7 B se realizó la simulación utilizando como sistema de apuestas el criterio de Kelly que tan buenos resultados dio en el experimento 7 A. Un resumen de los resultados de la evaluación es recogido en el *Anexo G: Resultados de las evaluaciones con el criterio de Kelly* pero si se quieren consultar los resultados con detalle estos se encuentran en los ficheros adjuntos en el CD del proyecto. Como ya se hizo en la anterior evaluación con este sistema se estableció al comienzo de cada uno de los meses una banca de 1000 unidades monetarias. Al comienzo de cada mes la banca volvía a estar en 1000 ya sea porque se hubiesen retirado beneficios o porque se aumentase porque se huebieran producido pérdidas. A continuación, se muestra el gráfico de la Tabla 50 con la evolución de los beneficios según el multiplicador aplicado y para cada una de las cuotas.

Como se ven los resultados para las apuestas a la cuota media tampoco fueron buenos con este sistema llegando a perder todos los meses prácticamente la banca entera en el caso de los multiplicadores altos.

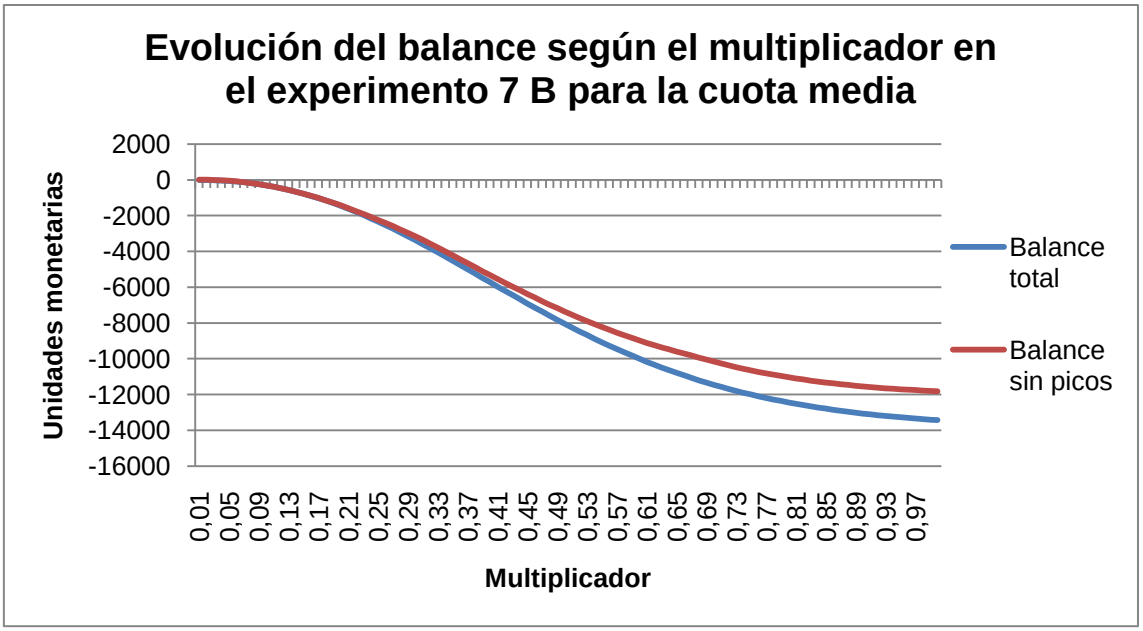


Tabla 50: Comparación del balance con los distintos multiplicadores utilizando como sistema de apuestas el criterio de Kelly en el experimento 7 B para la cuota media.

Los resultados obtenidos en el caso de apostar a la última cuota fueron positivos en un rango de valores del multiplicador por ello se presenta el gráfico de la Tabla 52Tabla 51 en el que se hace un *zoom* sobre ese rango para poder observar más claramente estos resultados.

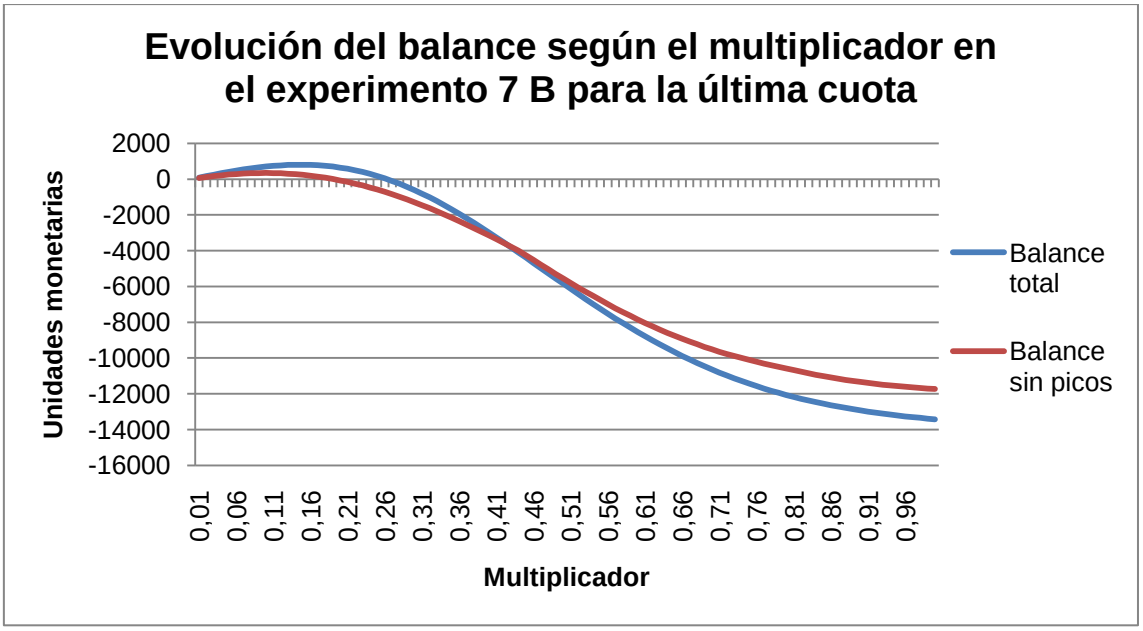


Tabla 51: Comparación del balance con los distintos multiplicadores utilizando como sistema de apuestas el criterio de Kelly en el experimento 7 B para la última cuota.

En el gráfico de la Tabla 52 se observa que con un multiplicador de 0,15 se superaron las 800 unidades monetarias de beneficio neto a lo largo de los 14 meses y en el caso de eliminar los resultados del mejor y peor mes se consiguieron unos beneficios de 345 unidades monetarias, por tanto, este sistema también obtuvo

resultados positivos en el caso de realizar apuestas a la última cuota registrada antes del comienzo de los partidos.

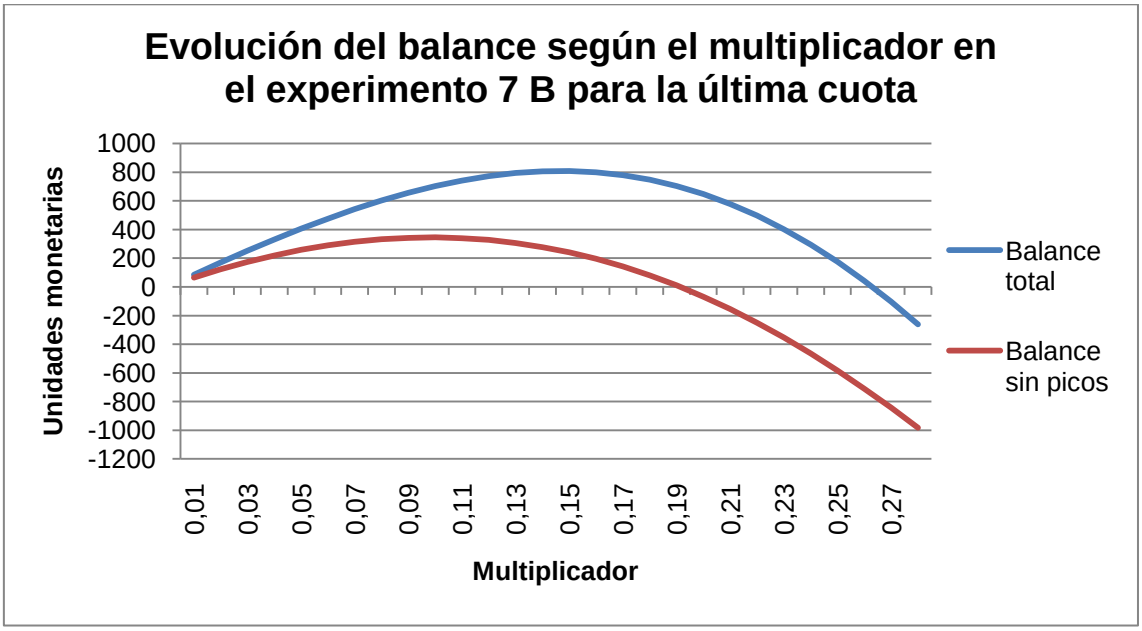


Tabla 52: Comparación del balance con los distintos multiplicadores que obtienen resultados positivos utilizando como sistema de apuestas el criterio de Kelly en el experimento 7 B para la última cuota.

Como en sistemas anteriores los resultados obtenidos para la cuota máxima fueron buenos, sin embargo, no eran fiables puesto que para obtener tan altos beneficios se tuvieron que realizar apuestas de gran importe, lo que planteaba problemas con el volumen de mercado y estos problemas no se podían tener en cuenta debido a que no se dispone de los datos necesarios. Los resultados siguen una gráfica similar a la conocida distribución normal maximizándose en los multiplicadores cercanos al 0,4. Estos resultados se muestran en el gráfico de la Tabla 53.

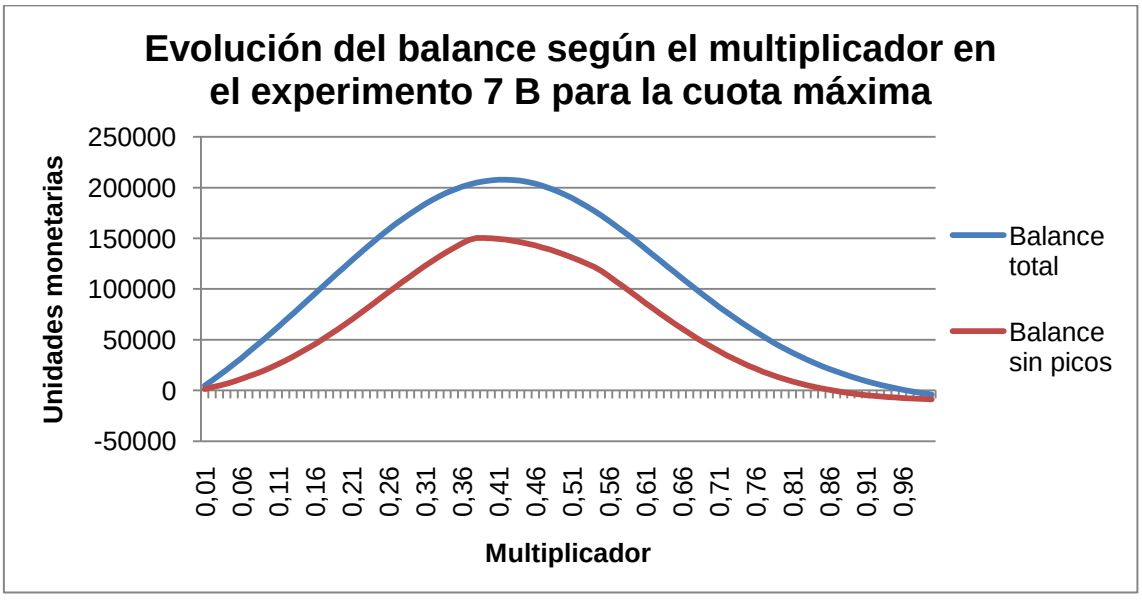


Tabla 53: Comparación del balance con los distintos multiplicadores utilizando como sistema de apuestas el criterio de Kelly en el experimento 7 B para la cuota máxima.

También, como en sistemas anteriores los resultados obtenidos apostando a la cuota mínima fueron en todos los casos negativos y peores a medida que aumentaba el multiplicador. A continuación se muestran en el gráfico de la Tabla 54:

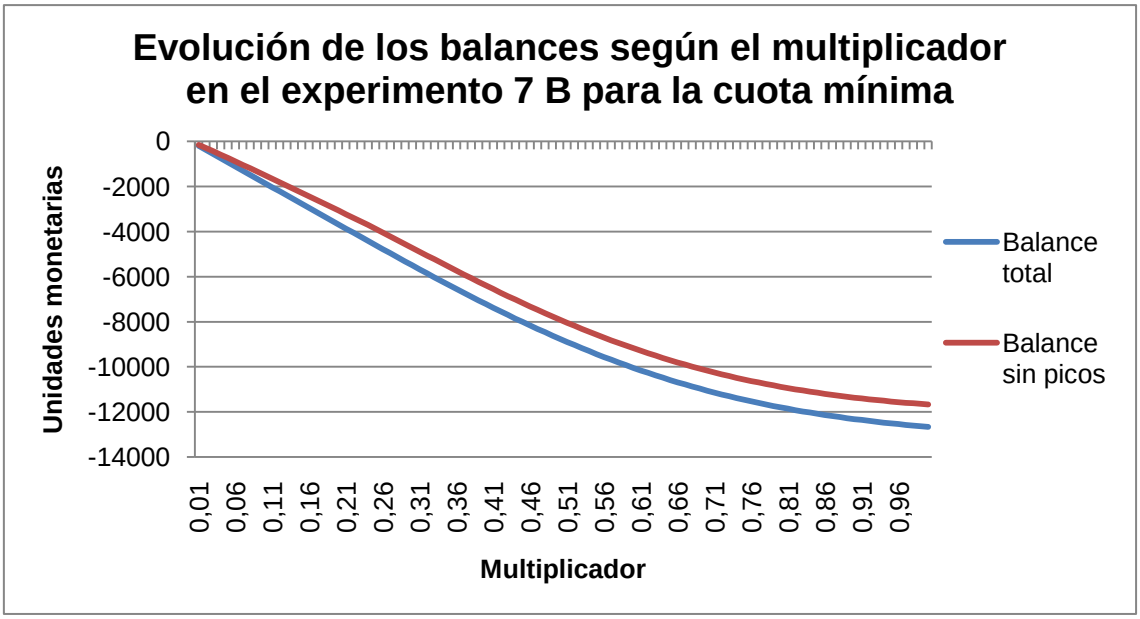


Tabla 54: Comparación del balance con los distintos multiplicadores utilizando como sistema de apuestas el criterio de Kelly en el experimento 7 B para la cuota máxima.

7.2.7.3 Evaluación del subconjunto C: Estadísticas enfrentadas con cuotas

Los pasos realizados para llevar a cabo la evaluación de este experimento fueron los mismos que los seguidos en el apartado *7.2.7.1*, esto fue así porque como ya se detalló en el Capítulo 6 los dos experimentos eran idénticos y la única diferencia entre ellos eran los datos con los que se generó el modelo.

El primer sistema de apuestas utilizado para evaluar el modelo es el sistema de apuestas fijas. Los resultados fueron peores que en el experimento 7 A generando unas pérdidas mayores. Sin embargo, las diferencias en cuanto a partidos en los que la predicción falló fueron mínimas puesto que en este caso sólo se falló la predicción en tres partidos más que en el experimento anterior. En la Tabla 55 se detallan estos resultados.

Resultados de la evaluación con el sistema de apuestas fijas	
Balance de inversiones a la última cuota	-35,95
Balance de inversiones eliminando el mejor y el peor mes	-32,48
Inversión (Número de partidos)	3.987
Predicción errónea (Número de partidos)	1.196

Tabla 55: Resultados de la evaluación del experimento 7 C con el sistema de apuestas fijas.

A continuación, se evaluó el modelo generado en este experimento con el sistema de apuestas basado en la cuota justa. En la Tabla 56 se detallan los resultados de la simulación del experimento.

Multiplicador	Balance	Balance sin picos	Rentabilidad	Invertido (Nº de partidos)	Predic. errónea
1	30,61	21,24	1,40%	2.191	708
1,01	37,2	29,62	1,84%	2.018	655
1,02	33,21	26,65	1,82%	1.826	603
1,03	20,02	12,77	1,21%	1.651	565
1,04	21,1	11,71	1,43%	1.480	517
1,05	32,16	24	2,48%	1.297	455
1,06	35,92	30,13	3,17%	1.134	405
1,07	35,38	30,82	3,53%	1.002	365
1,08	27,94	20,96	3,20%	872	327
1,09	18,65	12,34	2,43%	767	298
1,1	15,55	8,69	2,38%	654	263
1,11	11,99	6,9	2,14%	559	232
1,12	15,14	9,27	3,20%	473	200
1,13	24,57	16,34	6,04%	407	170
1,14	18,73	11,53	5,08%	369	160
1,15	8,99	3,65	2,79%	322	148
1,16	0,54	-0,86	0,20%	273	134
1,17	3,71	0,94	1,55%	239	118
1,18	5,02	3,02	2,33%	215	107
1,19	7,79	5,67	3,95%	197	98

Tabla 56: Resultados de la evaluación del experimento 7 C con el sistema de apuestas basado en la cuota justa.

El resultado de la evaluación del experimento con este sistema también fue peor que el obtenido en la evaluación realizada al experimento 7 A, en la cual se alcanzaron beneficios de hasta 55 unidades monetarias y en ningún caso generando pérdidas. Los mejores resultados en este caso también fueron obtenidos con el multiplicador tomando el valor 1,01 como se observa en el gráfico de la Tabla 57.

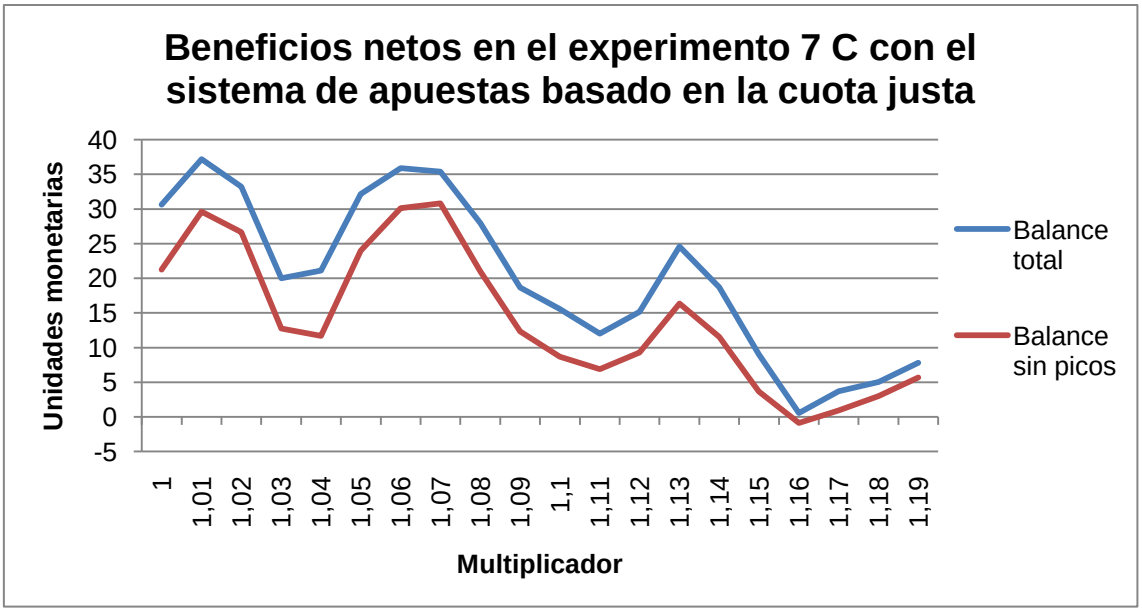


Tabla 57: Comparación del balance con los distintos multiplicadores en el sistema de apuestas basado en la cuota justa del experimento 7 C.

Al ser el resultado neto peor que en el experimento 7 A era de esperar que los resultados de la rentabilidad del modelo también fuesen peores. Sorprende sobre todo el cambio de tendencia que se produjo cuando el multiplicador tomó el valor 1,13 a partir del cual se produjo un cambio brusco en la rentabilidad hasta que el multiplicador tomó el valor 1,16 momento en el cual volvió a coger una tendencia positiva. Esta situación queda reflejada en el gráfico de la Tabla 58.

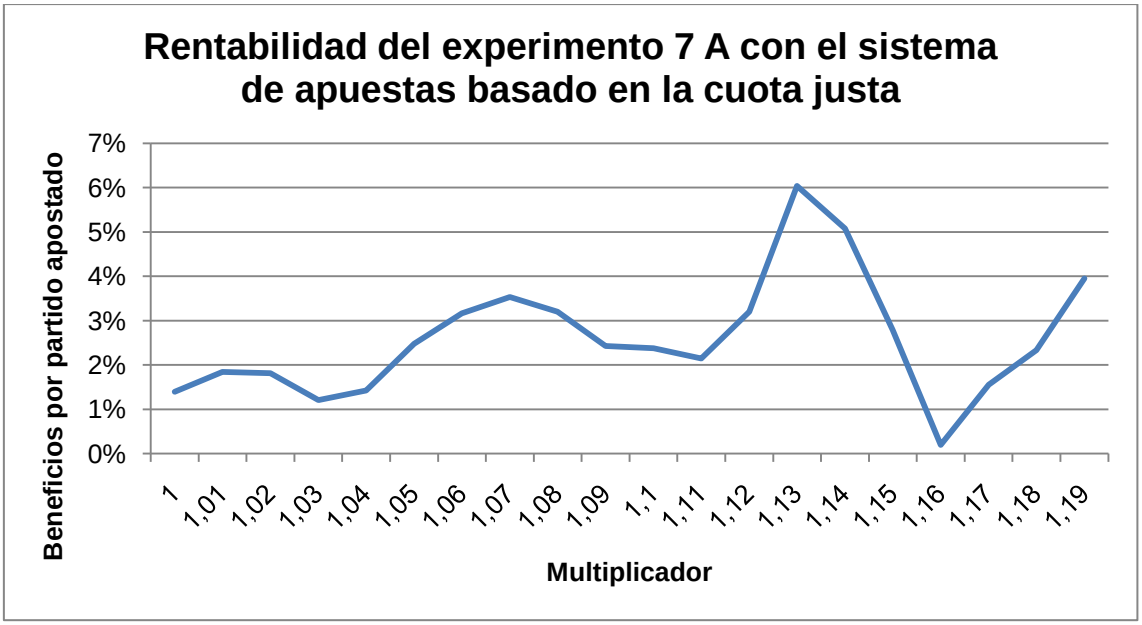


Tabla 58: Evolución de la rentabilidad en función del multiplicador aplicado en el experimento 7 C con el sistema de apuestas basado en la cuota justa.

El último sistema utilizado para evaluar el experimento fue el criterio de Kelly. La banca inicial, como en anteriores evaluaciones fue puesta a 1000 al comienzo de cada uno de los 14 meses.

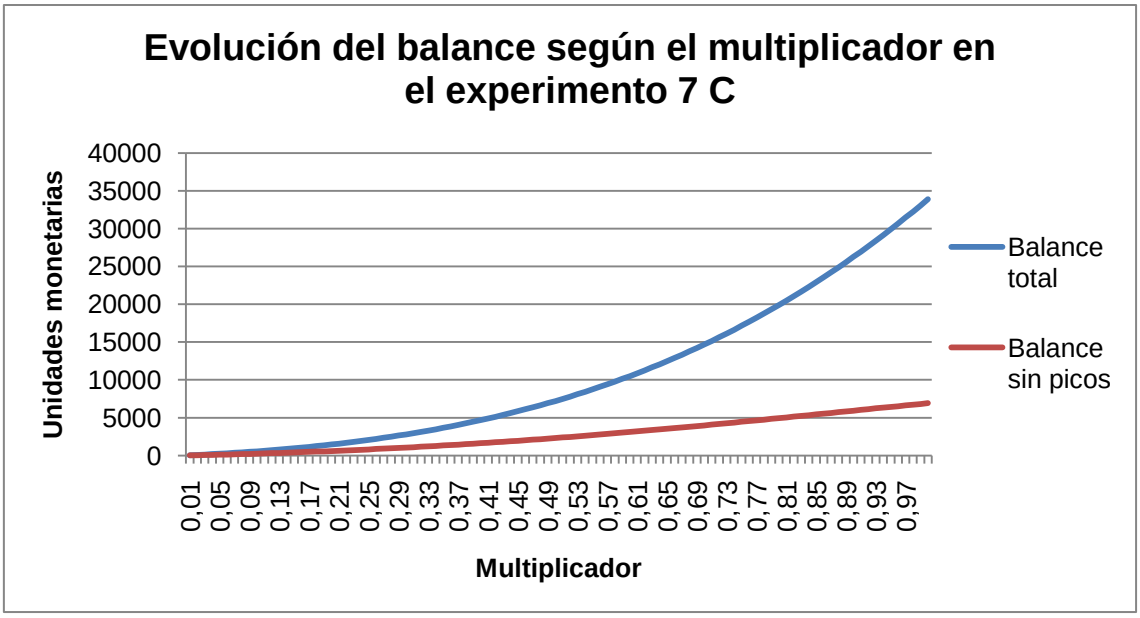


Tabla 59: Comparación del balance con los distintos multiplicadores utilizando como sistema de apuestas el criterio de Kelly en el experimento 7 C.

Como se puede observar en la Tabla 59 los resultados del balance total son realmente buenos llegando a obtener unos beneficios del 242% con el multiplicador tomando el valor 1, sin embargo, en el balance sin picos el beneficio neto bajó hasta los 6.916.52, es decir, más de 26.000, con lo que se obtuvieron unos beneficios del 57.53%. Esto fue debido a que en el mes de mayores ganancias, en este caso enero del 2009, se obtuvieron unas ganancias netas de más de 28.000, la explicación a los buenos resultados del mes es que se acertó la predicción en 62 partidos de los 83 en los que se apostó, o lo que es lo mismo se tuvo un porcentaje de acierto del 74,7%. Además, es muy posible que se encadenaran la mayoría de los aciertos seguidos y después de los fallos, de esta forma aumentaba la banca exponencialmente hasta obtener ese resultado. Otro de las posibles causas de la situación fue que se apostase a una cuota muy sobrevalorada debido a una equivocación de otro usuario.

7.2.7.4 Evaluación del subconjunto D: Estadísticas enfrentadas sin cuotas

La evaluación de este último experimento tenía que obtener mejores resultados que la del experimento 7 C ya que, el porcentaje de acierto obtenido fue más de un 2% mayor. El primer sistema utilizado volvió a ser el de apuestas fijas con el que se obtuvieron mejores resultados en los casos de que se apostó a la cuota máxima y a la cuota mínima, sin embargo, en caso de apostar a las cuotas media y última los resultados fueron peores como queda reflejado en la Tabla 60.

Descripción	Balance	Balance sin picos
Apostando a la cuota media	-68,32	-51,23

Descripción	Balance	Balance sin picos
Apostando a la cuota máxima	1.670,96	672,25
Apostando a la cuota mínima	-409,08	-348,13
Apostando a la última cuota	-47,1	-35,12
Inversión (Número de partidos)	3.987	3.450
Predicción errónea(Número de partidos)	1.340	1.169

Tabla 60: Resultados de la evaluación del experimento 7 D con el sistema de apuestas fijas.

El siguiente sistema con el que se evaluó el modelo es el basado en la cuota justa. Los resultados volvieron a ser peores que los obtenidos en el experimento 7 B para las cuotas media y última y mejoraron con las cuotas máxima y mínima. Sin embargo, los resultados con las cuotas máxima y mínima fueron menos relevantes que los obtenidos con las cuotas medias y última, por ser estos menos realistas, ya que, como se describió en otros apartados no se conocía en qué momento se iban a producir esas cuotas. Simplemente se daban esas medidas como el peor y el mejor de los resultados que podrían haber sido obtenidos en caso de apostar en el mejor y en el peor momento en todas las situaciones. Esta situación lógicamente es prácticamente imposible que ocurra ya que es muy difícil apostar en todos los partidos a la cuota máxima, aunque se hubiesen realizado estudios para predecir el instante en el que la cuota es máxima se conseguiría apostar en todos los partidos a esta cuota. La Tabla 61 refleja estos resultados.

Multiplicador	Balance cuota media	Balance cuota máxima	Balance cuota mínima	Balance última cuota
1	-38,71	1.643,7	-157,49	6,74
1,01	-39,94	1.640,02	-150,49	1,85
1,02	-47,19	1.636,02	-151,3	4,82
1,03	-53,22	1.625,97	-140,23	-6,66
1,04	-58,66	1.625,19	-131,12	-8,92
1,05	-62,29	1.612,54	-129,27	-13,8
1,06	-68,71	1.603,31	-130,93	-17,19
1,07	-64,83	1.588,99	-128,17	-17,94
1,08	-58,38	1.580,02	-125,11	-22,49
1,09	-57,9	1.571,79	-129,76	-23,82

Multiplicador	Balance cuota media	Balance cuota máxima	Balance cuota mínima	Balance última cuota
1,1	-51,41	1.575,2	-122,62	-20,06
1,11	-52,1	1.561,74	-110,94	-20,16
1,12	-57,48	1.565,34	-107,47	-30,37
1,13	-65,47	1.566,38	-101,87	-25,98
1,14	-63,58	1.566,26	-101,12	-19,47
1,15	-64,14	1.560,39	-100,72	-18,3
1,16	-52,6	1.555,54	-95,71	-17,93
1,17	-49,76	1.549,07	-96,41	-14,45
1,18	-46,66	1.543,93	-92,42	-8,2
1,19	-40,45	1.544,46	-93,4	-9,57

Tabla 61: Resultados de la evaluación del experimento 7 D con el sistema de apuestas basado en la cuota justa.

En el gráfico de la Tabla 62 se puede observar como el resultado de apostar a la cuota media generó en todo momento pérdidas siendo éstas menores con los multiplicadores más bajos y más altos. Esto fue así porque con los multiplicadores tomando valores bajos se apostaba más y también se acertaba más por lo que había una compensación y con los multiplicadores altos se apostaba menos, porque las condiciones para apostar se endurecían, y por lo tanto también se perdía menos.

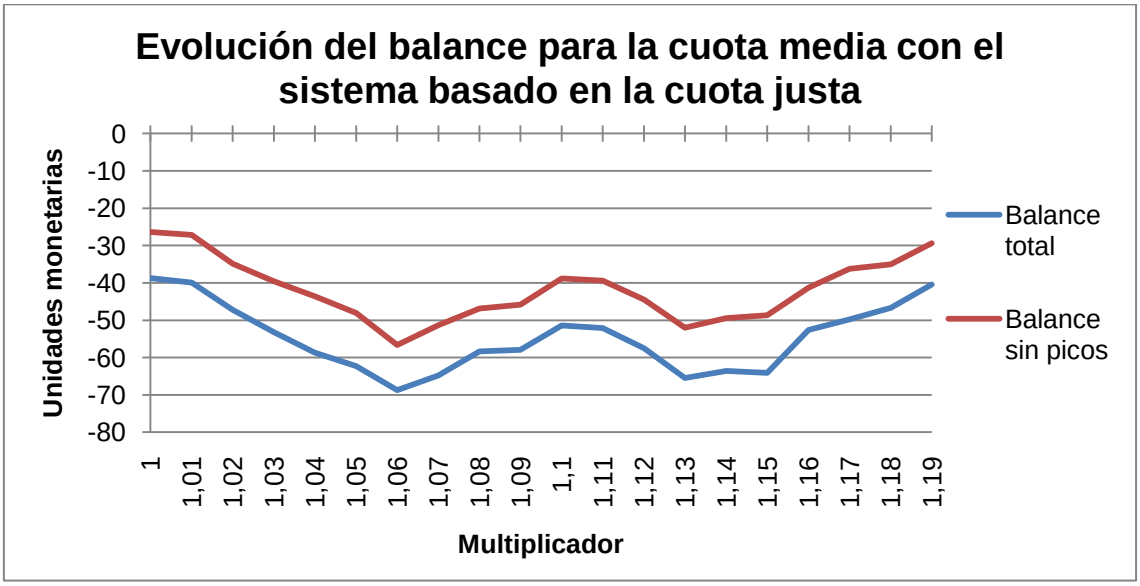


Tabla 62: Comparación del balance con los distintos multiplicadores con el sistema de apuestas basado en la cuota justa del experimento 7 D y con las apuestas realizadas a la cuota media.

En este caso los resultados obtenidos con la cuota máxima fueron mejores que en el experimento 7 B mejorando sobre todo el balance sin picos, esto fue debido a que el mejor mes en este experimento no fue mejor que el mejor del experimento 7 B. Sin embargo, en este caso se obtuvieron mejores resultados en términos generales de ahí la diferencia. Como se puede observar en el gráfico de la Tabla 63 el balance sigue una tendencia descendente a medida que el valor del multiplicador aumenta, esto fue debido a que como esta simulación contemplaba que la apuesta se realizaba cuando la cuota estaba más alta, normalmente siempre estaba muy por encima de su valor justo, es por ello que en cuantos más partidos se apostaba más se ganaba.

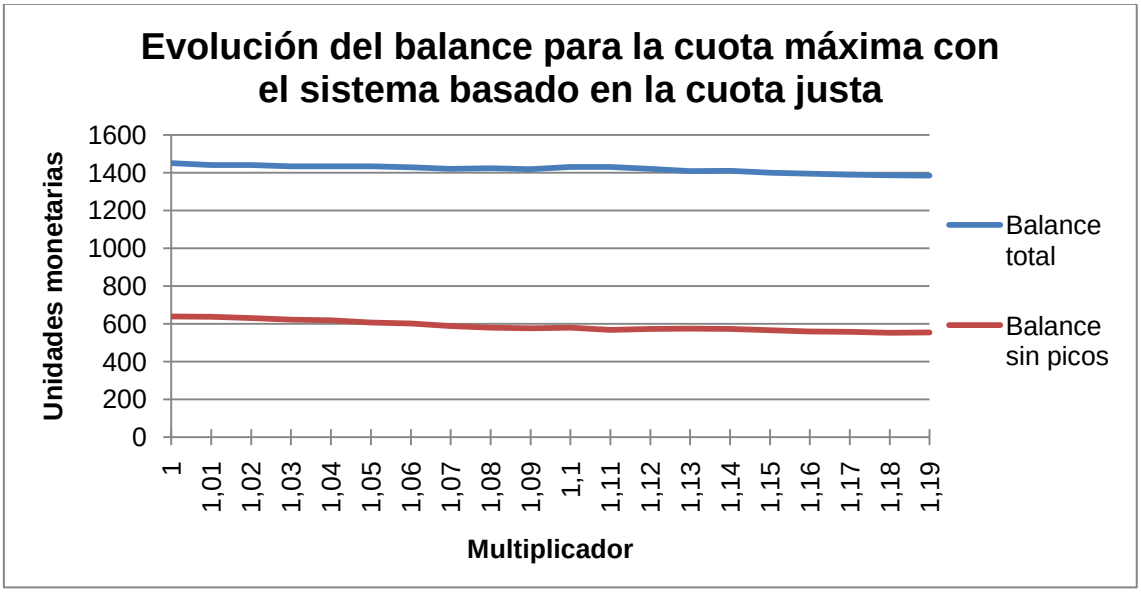


Tabla 63: Comparación del balance con los distintos multiplicadores con el sistema de apuestas basado en la cuota justa del experimento 7 D y con las apuestas realizadas a la cuota máxima.

Al apostar a la última cuota registrada antes del inicio de los partidos y con multiplicadores bajos se consiguieron resultados positivos pero muy por debajo de los conseguidos en el experimento 7 B como se puede observar en el gráfico de la Tabla 64.

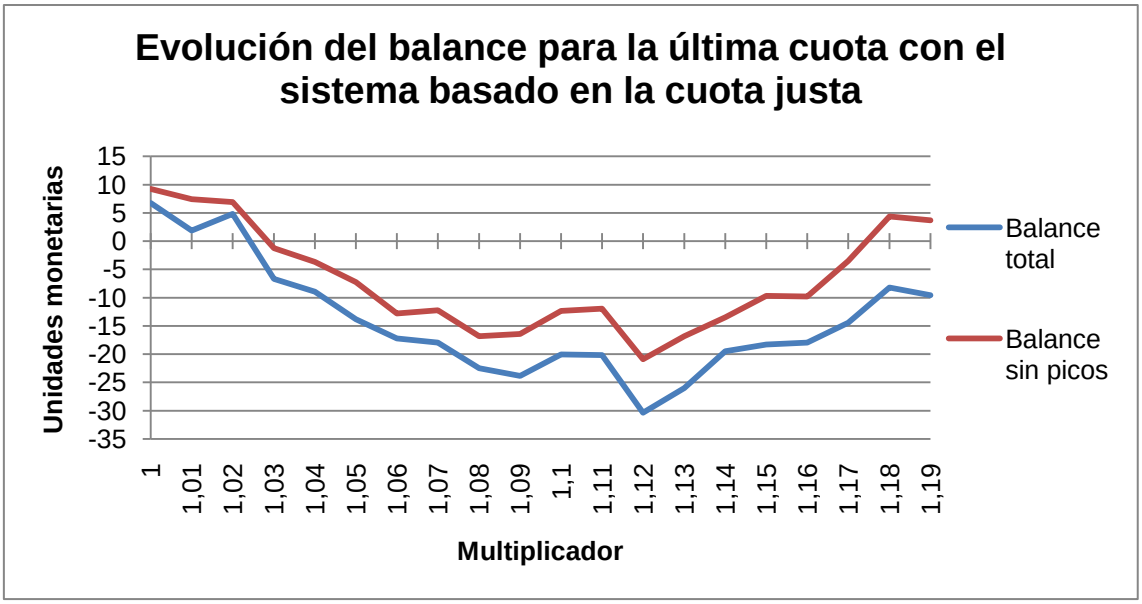


Tabla 64: Comparación del balance con los distintos multiplicadores con el sistema de apuestas basado en la cuota justa del experimento 7 D y con las apuestas realizadas a la última cuota.

Los dos gráficos siguientes muestran la rentabilidad obtenida al aplicar este sistema de apuestas al modelo generado en el experimento 7 D. En el gráfico de la Tabla 65 sólo se aprecia con claridad que al apostar a la cuota máxima se obtuvieron rentabilidades muy altas y al apostar a la cuota mínima la rentabilidad fue negativa.

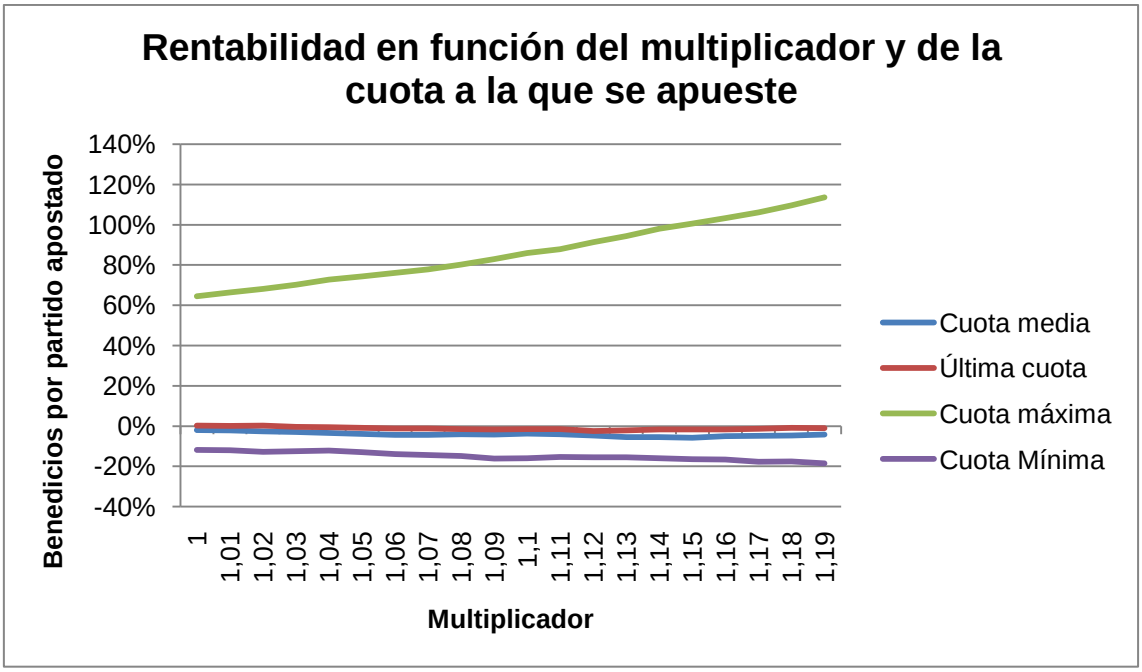


Tabla 65: Evolución de la rentabilidad en función del multiplicador aplicado en el experimento 7 D con el sistema de apuestas basado en la cuota justa.

En el gráfico de la Tabla 66 están reflejados los resultados de apostar a la última cuota y a la cuota media en el experimento 7 D, en este caso se puede observar que sólo se obtuvieron resultados con rentabilidad positiva en el caso de apostar a la última cuota y con el multiplicador siempre inferior a 1.02.

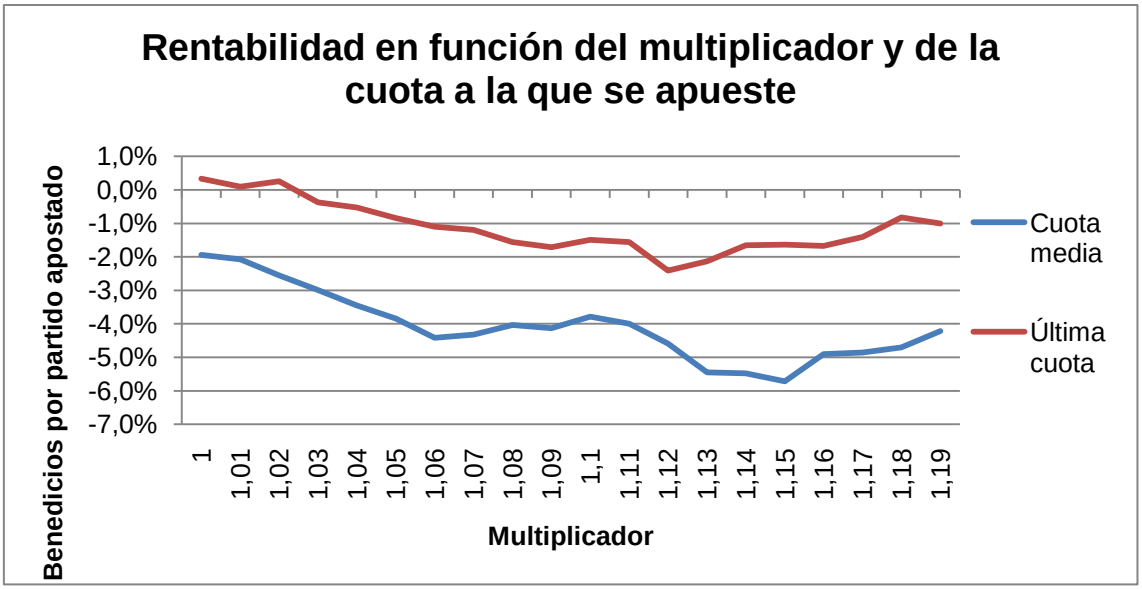


Tabla 66: Evolución de la rentabilidad en función del multiplicador en el experimento 7 D con el sistema de apuestas basado en la cuota justa para apuestas a la última cuota y a la cuota media.

La conclusión final que se pudo extraer de la evaluación del modelo generado en este experimento fue que siendo los porcentajes de acierto de la experimentación superiores a los conseguidos en el experimento similar, 7 B, la rentabilidad obtenida al aplicar el modelo generado por el experimento fue inferior a la obtenida por el modelo generado en el experimento 7 B.

El único caso que mostró resultados realmente rentables, fue apostando a la cuota máxima, sin embargo, como ya se ha mencionado en evaluaciones anteriores, la evaluación con la cuota máxima no es realista ya que no se podía saber de antemano en qué momento se iba a producir este pico ni tan si quiera qué valor iba a llegar a tomar.

7.3 Evaluación final

En este apartado se muestra un resumen de los resultados más relevantes de la evaluación. Además se realizan comparativas entre dichos resultados. Los únicos modelos interesantes desde el punto de vista de la rentabilidad económica fueron los generados en el Experimento 7. En este experimento se crearon cuatro modelos distintos en dos de los cuales se utilizaron las cuotas para su creación, mientras que en los otros dos éstas fueron ignoradas.

En los modelos en los que las cuotas no fueron utilizadas se realizó la evaluación con cuatro tipos de cuotas distintas, sin embargo, en esta comparativa final sólo se tuvo en cuenta la última cuota registrada por tres motivos. El primer motivo fue por ser la cuota con la que se evaluaron los modelos creados con información de las cuotas, el segundo motivo, ya ha sido expuesto anteriormente, es que la evaluación

con esta cuota era la más real y el tercer motivo fue que en caso de utilizar el modelo en un futuro sería la cuota más sencilla de encontrar ya que simplemente habría que esperar al instante anterior al comienzo de un partido para apostar.

En gráfico de la Tabla 67 se muestran los resultados obtenidos con el sistema de apuestas fijas en el experimento 7. En este caso los resultados fueron siempre negativos por lo que se pudo deducir que, el sistema de apuestas, el modelo creado o ambos no eran buenos. El modelo que menores pérdidas generó fue el obtenido en el experimento 7 A que utilizaba las estadísticas completas y separadas y los datos de las cuotas conjuntamente.

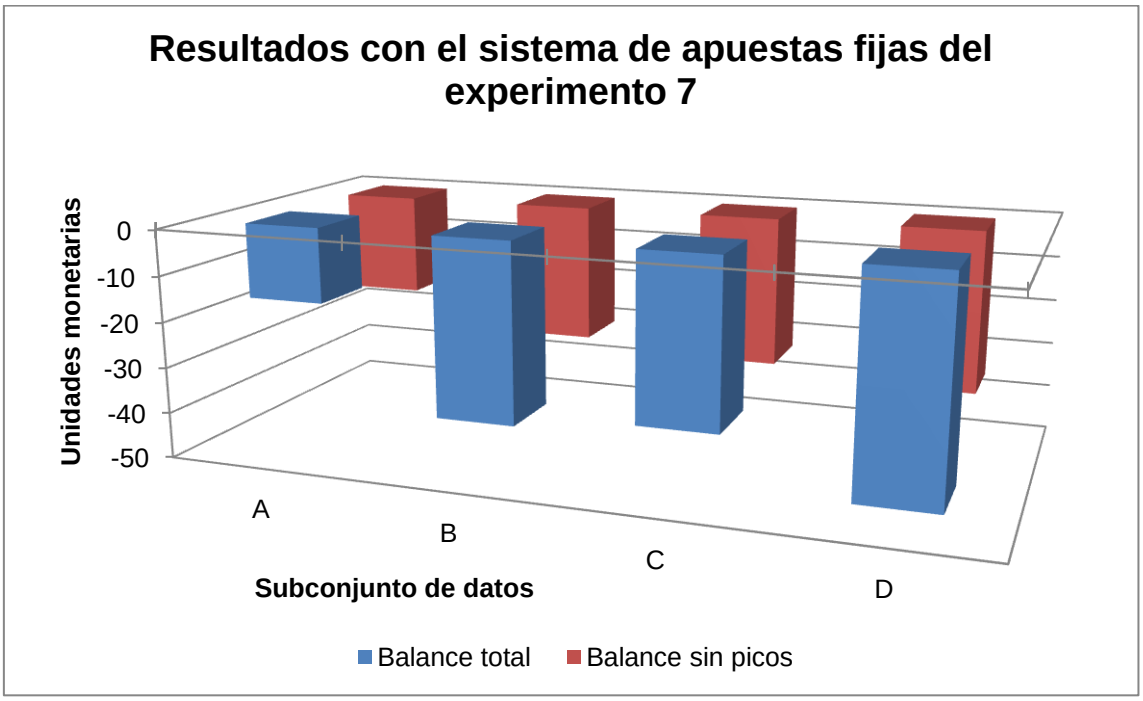


Tabla 67: Comparativa de resultados de la evaluación con el sistema de apuestas fijas del experimento 7.

A continuación, en el gráfico de la Tabla 68 se ofrece una comparativa de los resultados obtenidos utilizando el sistema de apuestas basado en la cuota justa. En todos los casos, a excepción del modelo generado con el subconjunto de datos D, se produjeron resultados positivos. El mejor modelo generado para este sistema de apuestas fue el generado con el subconjunto de datos A, con el cual se llegaron a obtener unos beneficios de 54,08 unidades monetarias en el mejor de los casos teniendo en cuenta únicamente el beneficio total. Si lo que se tiene en cuenta es la rentabilidad de las inversiones, el mejor modelo generado también fue el generado con el subconjunto de datos A llegando a obtener una rentabilidad del 11.72%.

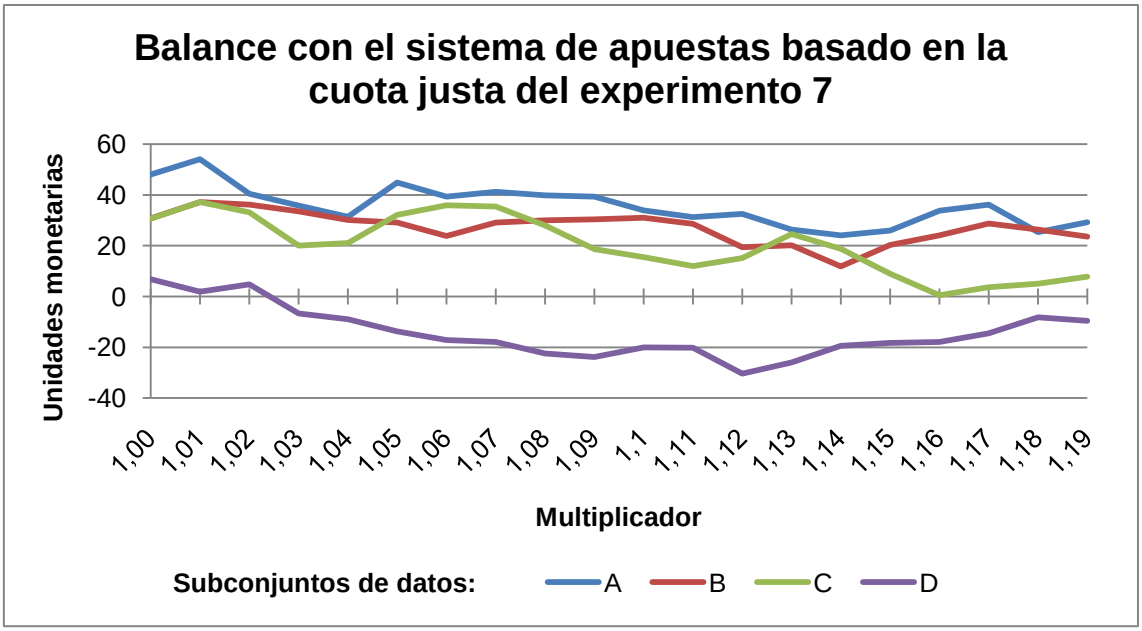


Tabla 68: Resultados del balance obtenido apostando a la última cuota con el sistema de apuestas basado en la cuota justa del experimento 7.

Si se observan las dos gráficas se aprecia claramente como la gráfica de la Tabla 68 (balance) tiene una pendiente descendente mientras que la de la Tabla 69 (rentabilidad) es ascendente. Esto es debido a que los multiplicadores bajos favorecían que se realizase un mayor número de apuestas, mientras que los multiplicadores altos endurecían las condiciones de realizar las apuestas sólo realizando apuestas que estaban hasta un 19% mejor cotizadas que la cotización justa asignada por el modelo. La pregunta que se tendría que hacer un inversor al decidir qué multiplicador utilizar para llevar a cabo este sistema de apuestas es qué es mejor, ganar más dinero o aumentar la rentabilidad del dinero invertido.

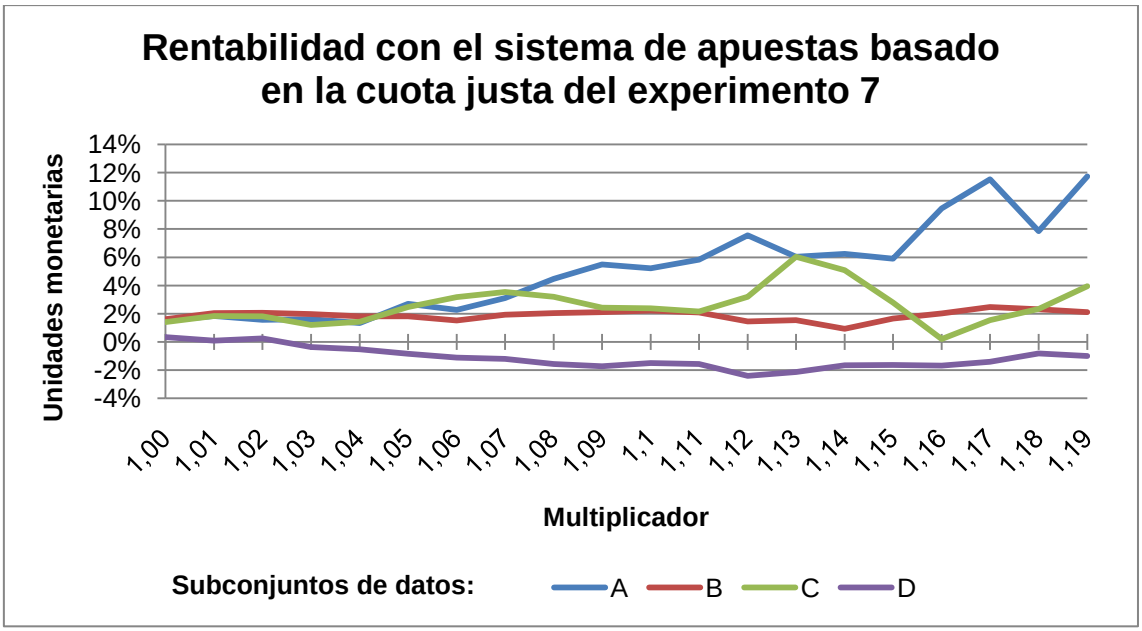


Tabla 69: Rentabilidad obtenida apostando a la última cuota con el sistema de apuestas basado en la cuota justa del experimento 7.

Por último, en el gráfico de la Tabla 70 se comparan los resultados obtenidos utilizando como sistema de apuesta el criterio de Kelly. En este caso no se muestra la gráfica con el balance total del experimento puesto que éste puede ser obtenido a través de los datos de rentabilidad ya que, la inversión fue la misma con todos los multiplicadores. Como se comentó en apartados anteriores se hizo una inversión de 1.000 unidades monetarias cada mes para un total de 14 meses que duró la simulación. Por tanto, la inversión final fue de 14.000 unidades monetarias y se terminó en el mejor de los casos con 47.879.09 unidades monetarias, es decir, una rentabilidad de más del 240%. En este caso, el modelo más rentable fue el generado con el subconjunto de datos C, que siguió una pendiente ascendente a medida que el multiplicador era mayor. El único modelo que generó pérdidas con cualquier valor que tomaba el multiplicador fue el modelo generado con el subconjunto de datos D.

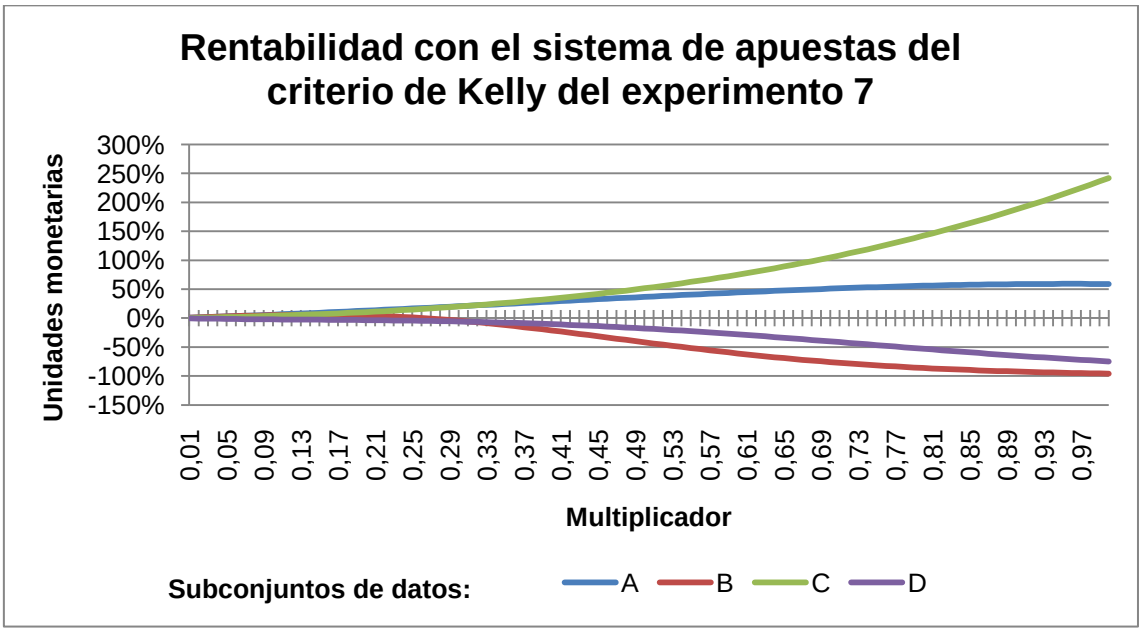


Tabla 70: Rentabilidad obtenida apostando a la última cuota con el sistema de apuestas del criterio de Kelly del experimento 7.

A continuación, en el gráfico de la Tabla 71 se muestra una apreciación de la gráfica de la Tabla 70 mostrando los resultados para los multiplicadores menores. Se recuerda que cuanto más alto es el valor del multiplicador mayor es el riesgo tomado en cada apuesta realizada, esto ocasiona que si un mes se ganan muchas apuestas se puede elevar en grandes cantidades el valor inicial de la banca, sin embargo, si en ese mes no se obtienen buenos resultados la banca, sin llegar nunca a quedarse a cero, toma valores muy cercanos. Con este sistema lo que se observó es que con los multiplicadores altos la mayoría de los meses la banca se perdía casi entera, mientras que, en los meses con ganancias estas pérdidas se compensaron con creces. Por ejemplo, en el caso del modelo generado con el subconjunto de datos C y con el multiplicador tomando el valor 1, es decir, el caso de mayor ganancia de todos, la banca acabó por encima de la inversión inicial de 1.000 únicamente en 5 de los 14 meses, sin embargo, las ganancias de esos 5 meses compensaron las pérdidas generadas en los otros 9.

En cualquier caso, hay que tener cuidado con estos resultados pues es muy probable que en los meses con grandes ganancias se tuvieran que hacer al final de mes apuestas de gran cantidad en los casos donde los multiplicadores tenían un valor alto. En estos casos, el usuario se podría quedar sin volumen de mercado para cubrir apuestas tan grandes. Estas situaciones no pudieron ser controladas en este proyecto, ya que, los datos de *Betfair* no tenían información del volumen de mercado en cada instante. Por tanto, ante esta falta de control, los resultados más fiables fueron los obtenidos para los multiplicadores con valores más pequeños.

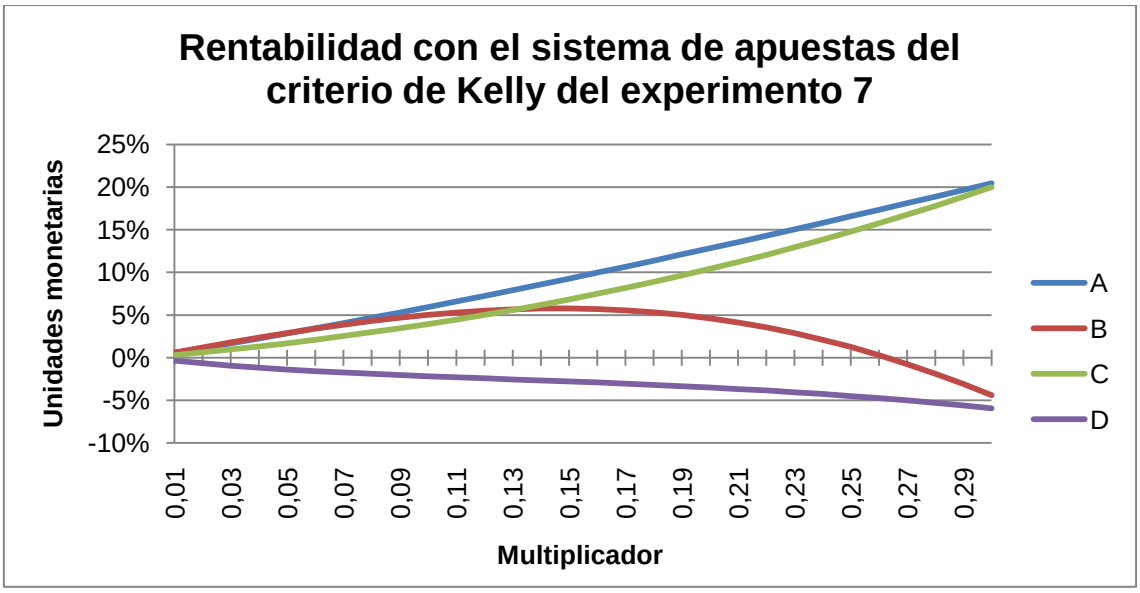


Tabla 71: Rentabilidad obtenida apostando a la última cuota por los multiplicadores bajos con el sistema de apuestas del criterio de Kelly del experimento 7.

A continuación, en la Tabla 72 se muestran los resultados obtenidos para cada uno de los meses con el modelo B y el multiplicador tomando el valor 0.15.

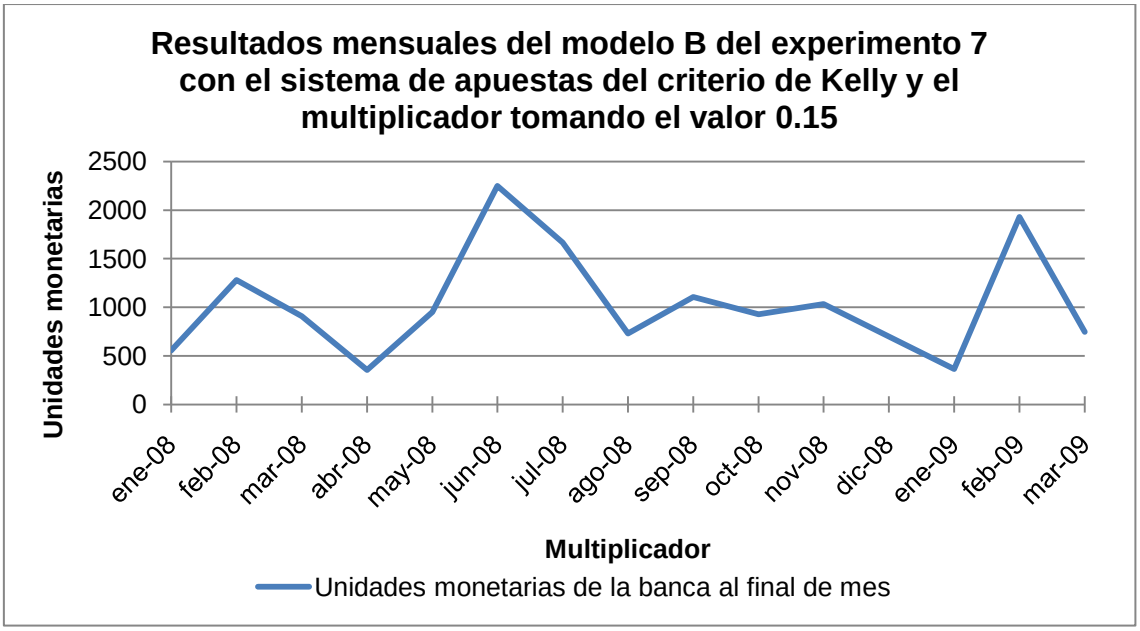


Tabla 72: Resultados mensuales del modelo B del experimento 7 con el sistema de apuestas del criterio de Kelly y el multiplicador tomando el valor 0.15.

Como se puede observar en seis de los meses se consiguieron resultados positivos mientras que en 8 meses se producieron pérdidas. En el mejor mes la banca aumentó más de un 100% mientras que en el peor mes se perdió más del 60% de la banca.

Por tanto, una vez analizados todos los resultados se está en disposición de afirmar que se han encontrado modelos ganadores para los sistemas de apuestas basados tanto en la cuota justa como en el criterio de Kelly, los mejores modelos creados son los generado con el subconjunto de datos A y con el subconjunto de datos C difiriendo sus resultados según el sistema de apuestas empleado. Además, el criterio de Kelly es el sistema de apuestas que obtuvo mayor rentabilidad.

Capítulo 8: DESPLIEGUE

En este capítulo se determinarán los pasos necesarios para la aplicación del estudio realizado en el proyecto.

El primer paso será la actualización de la base de datos con la que se ha trabajado a lo largo del proyecto. Esta base de datos deberá de ser actualizada todos los meses para posteriormente generar el modelo de predicción del siguiente mes. Para llevar a cabo esta tarea sería conveniente la creación de un programa sencillo que inserte los datos obtenidos de *OnCourt* y *Betfair* en la base de datos, de esta forma, se automatizaría el proceso con lo que se realizaría la tarea más rápidamente.

Una vez se haya actualizado la base de datos el siguiente paso es la creación del modelo de predicción. Para esto se necesitará crear el fichero arff. También sería conveniente la implementación de otro programa que automatizara la tarea. Una vez se dispusiese del fichero, éste se introduciría en Weka y se crearía el modelo de predicción utilizando el algoritmo `DecisionTable` con el método de búsqueda `LinearForwardSelection` que como ya se ha estudiado a lo largo del proyecto es con el que mejores resultados se obtuvieron.

Una vez creado el modelo de predicción se tomarían los datos del partido en el que se quisiese apostar, se aplicaría el modelo a estos datos y se obtendría una predicción. Esta predicción tiene un porcentaje de acierto que habría que calcular. Para obtener el porcentaje de acierto habría que seleccionar todas las instancias del fichero de entrenamiento, con el que se generó el modelo, que cumplan las condiciones que tiene el partido a predecir. Es decir, todos los ejemplos que cumplan las mismas reglas que se utilizan para predecir el partido. Una vez seleccionadas se analizan en cuantas instancias se acertó o falló la predicción y se calcula la probabilidad de acierto, que será necesaria a la hora de aplicar el sistema de apuestas.

Con la predicción obtenida por el modelo se elegiría el sistema de apuestas a emplear. Por ejemplo, el criterio de Kelly con el cual se obtuvieron los mejores resultados. Se aplicaría la fórmula de este criterio, la cual está expuesta en el apartado *7.1.3 El criterio de Kelly*, y se llevaría a cabo la apuesta.

Evidentemente un paso previo a todos los anteriores sería el de tener una cuenta en la casa de apuestas *Betfair*. Así pues, uno de los primeros pasos sería el registro en la casa de apuestas, este registro puede realizarse a través de la página web de la compañía[1]. Una vez realizado el registro habría que estudiar el dinero disponible para invertir y dividirlo en un número suficiente de partes para garantizar el éxito, se recomienda por ejemplo su división en 12 partes con lo que se cubriría una temporada completa de tenis. De esta forma se aseguraría el poder reiniciar la banca durante un año y no correr el riesgo de bancarrota en caso de que los primeros meses no se obtuvieran resultados positivos.

Una vez seguidos estos pasos ya se estaría en disposición de aplicar el modelo durante un mes. Finalizado el mes se debería de actualizar la base de datos y generar un nuevo modelo. El estudio realizado en este proyecto ha llevado a cabo una simulación actualizando la base de datos y creando un nuevo modelo todos los meses. Sin embargo es muy posible que los resultados mejorasen si se acortase este tiempo y por ejemplo, se actualizase la base de datos y se generase un nuevo modelo cada semana.

[1] https://account.betfair.com/account-web/registerAccount.html?origin=GOAALL

Capítulo 9: CONCLUSIONES Y TRABAJOS FUTUROS

Con la realización de este proyecto de fin de carrera se pretendió aplicar técnicas de minería de datos para la predicción de resultados en eventos deportivos. Estas técnicas han tenido gran éxito en distintos ámbitos como en la medicina, la banca, el marketing, los procesos industriales, etc.

Determinar el ganador de un evento deportivo no es nada sencillo, entran en juego distintos factores que se escapan de cualquier análisis lógico o estadístico. En este caso el proyecto fue enfocado sólo a la predicción de los resultados de un deporte y de una modalidad de ese deporte Se trató de predecir el ganador de un partido de tenis masculino e individual. Cada partido era disputado por dos únicos jugadores de los cuales se disponía de información de sus estadísticas de juego, también eran conocidos datos del partido como el lugar, el torneo, la ronda, el tipo de superficie, etc. Sin embargo, se desconocían otros muchos factores como el estado anímico de cada jugador, su situación personal, su estado físico, etc. El deporte profesional al más alto nivel está muy igualado, cientos de jugadores profesionales compiten por ser los mejores y ganarse la vida sobre las pistas de tenis. Es evidente que muchos de ellos llegan a ser millonarios pero otros muchos tienen que vivir de las ganancias obtenidas durante su carrera deportiva como profesionales, que no suele durar más de 15 años. Por tanto, cada partido se convierte en una batalla en la que, en muchos casos, se decide el ganador por simples detalles. Estos detalles son los que dificultaron la tarea de predecir el resultado final del encuentro.

9.1 Conclusiones

De ante mano se conocía la dificultad que llevaría conseguir un modelo que obtuviese porcentajes de aciertos cercanos al 100%. En este proyecto el mejor

resultado obtenido no superó el 71% de acierto. Sin embargo, aunque este bajo porcentaje pudiera indicar lo contrario, si permitió realizar inversiones en el mercado de las apuestas con rentabilidad positiva.

La tarea más costosa a lo largo del proyecto fue la preparación de los datos. Por un lado, los datos fueron obtenidos de dos fuentes distintas con lo que tenían distintos formatos y distintas formas de representar los datos. Por ejemplo el tenista Rafael Nadal era representado en una fuente como "Rafael Nadal Parera" mientras que en la otra fuente en ocasiones era representado sólo como "Nadal", otras como "R. Nadal", "Rafa Nadal" o "Rafael Nadal". Esto mismo ocurría con otros jugadores y con otros atributos totalmente distintos. Además, una de las fuentes de datos, *Betfair*, también iba variando el formato de sus datos así como la representación de los mismos a lo largo de los años. Por tanto, ésta fue la tarea más costosa del proyecto ya que hubo que dedicarle mucho tiempo para poder conseguir unos resultados útiles para las siguientes tareas.

Se han probado distintos algoritmos de clasificación y una de las conclusiones que se obtiene del análisis de los resultados obtenidos es que en este caso los algoritmos que obtenían unos modelos más sencillos son los que mejores resultados han obtenido. A lo largo del modelado se han utilizado desde árboles de decisión hasta algoritmos bayesianos obteniendo los mejores resultados con los algoritmos basados en reglas, los que a su vez generaban los modelos más simples.

Otro de los aspectos analizados a lo largo del proyecto fue que el beneficio obtenido al aplicar algoritmos de selección de atributos variaba de unos algoritmos a otros obteniendo resultados idénticos, mejores o peores según el caso. Por tanto, se concluyó que está técnica debía de ser estudiada en cada caso, ya que había posibilidades de obtener buenos resultados con su aplicación. Una de las claras ventajas obtenidas con esta técnica fue la disminución del tiempo requerido para las tareas de clasificación al utilizar únicamente los mejores atributos indicados por las técnicas de selección de atributos. De esta forma, habría que estudiar si es o no rentable la utilización de este tipo de técnicas en cada caso.

A través de la evaluación de los modelos se consiguió demostrar que es posible obtener rentabilidad de una posible inversión en el mundo de las apuestas. En la casa de apuestas *Betfair*, en este caso más concretamente casa de intercambio de apuestas, se realizan en momentos de hora punta más de 300 transacciones por segundo, se cruzan más de un millón de apuestas semanales facturando más de 50 millones de euros a la semana y su lista de usuarios asciende a más de dos millones de personas. Por tanto, para obtener rentabilidad positiva el modelo de predicción generado debe ser mejor que el de resto de usuarios (en caso de que usen algún modelo que en la menor parte de los casos será lo que ocurra). *Betfair* es mundialmente conocida por su modelo de intercambio de apuestas, si se apostase en una casa de apuestas común el modelo generado tendría que ser mejor que el de la casa de apuestas, esto complicaría la obtención de beneficios puesto que las casas de apuestas tienen sus propios profesionales encargados de la tasación de las cuotas o de la generación de modelos que generen esas cuotas. Sin embargo, al competir contra otros usuarios, muchos de ellos inexpertos que basan sus apuestas en

sentimientos o intuiciones, hay un margen muy amplio para obtener beneficios a largo plazo.

Como se ha podido constatar con la experimentación llevada a cabo en el proyecto, el tiempo dedicado a la aplicación de los algoritmos de clasificación es muy elevado cuando se trabaja con grandes cantidades de datos, es por ello que no se pudo experimentar con distintas configuraciones de los parámetros de cada uno de los distintos algoritmos de clasificación utilizados. Es posible que con modificaciones en algunos de los parámetros se hubieran conseguido mejoras en los resultados, sin embargo, el tiempo requerido hubiera sido mucho mayor y en este caso se decidió experimentar con la configuración ofrecida por defecto por el software Weka.

Con la elaboración del proyecto se ha aprendido a seguir la metodología CRISP-DM en un proyecto. Cabe resaltar la necesidad de retornar e iterar continuamente entre las distintas fases, realizando muchos experimentos para acercarse lo más posible a los objetivos buscados.

Como conclusión final hay que destacar que se consiguieron todos los objetivos marcados al inicio del proyecto. El primero de ellos era la aplicación de técnicas de inteligencia artificial las cuales fueron utilizadas a través del software Weka y los algoritmos de clasificación y de selección de atributos que éste implementa. También se consiguió saber cuáles son los atributos más determinantes a la hora de determinar quién tiene más probabilidades de ser el ganador de un partido. Además, utilizando los movimientos de las apuestas se consiguió mejorar el porcentaje de acierto en la predicción del ganador de los partidos. Finalmente se elaboraron distintos modelos y sistemas con los que se pudo maximizar las ganancias obtenidas de una posible inversión en el mercado de las apuestas deportivas.

9.2 Trabajos futuros

Muchas son las propuestas que se pueden plantear para la continuación o mejora de este proyecto fin de carrera, a continuación se exponen algunas de las mismas:

- Trabajando con los mismos datos de este proyecto se podría plantear la experimentación de una forma distinta y mucho más costosa, tanto en tiempo como en recursos hardware, pero con la que se podrían mejorar los resultados. En vez de trabajar con un único fichero de datos con todos los partidos de tenis juntos, se podría formar para cada partido un fichero de datos distinto en el que se introducirían únicamente datos de partidos anteriores de los dos jugadores que disputan el partido que se quiere predecir. Esto supondría que para cada uno de los partidos a predecir habría un fichero de datos distinto, al que habría que aplicar el algoritmo de clasificación que ofreciera mejores resultados. Este algoritmo sería seleccionado después de un proceso de experimentación como el

seguido a lo largo de este proyecto. De esta forma se generaría un modelo de predicción para cada partido por lo que el modelo sería mucho más específico que el obtenido en este proyecto que es único para todos los partidos.

- Los resultados obtenidos en la evaluación de este proyecto podrían ser mejorados haciendo un estudio para predecir en qué momento se producen los picos de la cuota a favor de cada jugador. De esta forma la simulación realizada en la que se apostaba a la cuota máxima sería más real ya que gracias a este estudio se podría predecir el momento en el que se produce esta cuota.

- Una posible continuación del proyecto es ampliar el estudio a otras modalidades del tenis como el tenis masculino en modalidad de dobles, el tenis femenino y el tenis mixto en las modalidades de dobles e individual.

- Dentro del tenis también se podría ampliar el estudio de forma que no se intentase predecir únicamente el ganador del partido sino que se estudiasen otros tipos de apuestas como apuestas por sets, apuestas juego a juego, número de sets, apuestas a *tie break*, apuestas con hándicap, número de juegos, etc.

- El mundo de las apuestas mueve millones de euros en todo el mundo y en todos los deportes por lo que estudios de este tipo podrían ser realizados para cualquiera de los deportes disponibles en el mercado de apuestas. Estos deportes van desde los más populares como el fútbol o el baloncesto hasta deportes mucho menos conocidos en nuestra sociedad como los dardos, el beisbol, el criquet o el fútbol australiano.

GLOSARIO DE ACRÓNIMOS

AEDAPI: Asociación Española de Apostadores por Internet

ATP: Asociación de Tenistas Profesionales

DM: Data Mining. Minería de datos

ESPN: Entertainment & Sports Programming Network

IA: Inteligencia Artificial

IDE: Integrated Development Environment. Entorno de Desarrollo Integrado.

PFC: Proyecto Fin de Carrera

WTA: Women's Tennis Association

BIBLIOGRAFÍA

Betfair. (n.d.). *Betfair*. Retrieved Marzo 2010, from Betfair: http://www.betfair.com/es/aboutUs/

CRoss Industry Standard Process. (1996). *CRISP-DM*. Retrieved Marzo 16, 2010, from CRISP-DM: http://www.crisp-dm.org

Dietterich, T. G. (2000). Proceedings of the First International Workshop on Multiple Classifier Systems. In T. G. Dietterich, *Ensemble Methods in Machine Learning.* Berlin: Springer-Verlag London, UK.

Dietterich, T. G. (2000). *Ensemble Methods in Machine Learning.* Berlin: Springer-Verlag London, UK.

Dietterich, T. G. (2000). *Ensemble Methods in Machine Learning.* Berlin: Springer-Verlag London, UK.

Dietterich, T. G. (1997). *Machine-learning research: Four current directions.* AI Magazine.

Eclipse. (2001, 11). *Eclipse*. Retrieved 03 22, 2010, from Eclipse: http://www.eclipse.org/

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). In U. M. Fayyad, G. Piatetsky-Shapiro, & P. Smyth, *From data mining to knowledge discovery: an overview* (pp. 37-54). Al Magazine.

Hall, M. A. (1998). *Correlation-based Feature Selection for Machine Learning.* Hamilton, Nueva Zelanda.

Hall, M. (n.d.). *Weka.* Retrieved Marzo 2010, from Weka: http://www.kdd.org/explorations/issues/11-1-2009-07/p2V11n1.pdf

Hart, R. D. (1973). *Pattern classification and scene analysis.* New York: John Wiley & Sons.

Holte, R. C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. In R. C. Holte, *Machine Learning* (pp. 11: 63-91). Ottawa.

Kohavi, R. (1995). *The Power of Decision Tables.* Londres: Springer-Verlag.

Lorenzo. (2002). *Selección de Atributos en Aprendizaje Automático basado en la Teoría de la Información.* Gran Canaria.

Matjaž Gams, M. B. (1994). *A schema for using multiple knowledge.* Berkeley, California: MIT Press, Cambridge, MA, USA.

Meir R, R. G. (2003). *An Introduction to Boosting and Leveraging.* Springer.

OnCourt. (n.d.). *http://www.oncourt.info*. Retrieved Marzo 2010, 26, from http://www.oncourt.info: http://www.oncourt.info/index.html

Peter Buhlmann, B. Y. (2002). Analyzing bagging. In B. Y. Peter Buhlmann, *The Annals of Statistics* (pp. 30:927-61). Zurich y California: Institute of Mathematical Statistics.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* San Mateo: Morgan Kaufmann Publishers.

Tapia, F. F. (n.d.). *Red Científica: Ciencia, Tecnología y Pensamiento*. Retrieved Marzo 2010, from Inteligencia Artificial: http://www.redcientifica.com/gaia/ia/intia_c.htm

Wikipedia. (2001, Enero 15). *Wikipedia*. Retrieved Marzo 16, 2010, from Wikipedia: http://es.wikipedia.org/wiki/Inteligencia_artificial

Wikipedia. (2001, 01 15). *Wikipedia*. Retrieved 03 22, 2010, from Wikipedia: http://es.wikipedia.org/wiki/Weka_(aprendizaje_autom%C3%A1tico)

Witten I., F. E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* San Mateo: Morgan Kaufmann.

ANEXOS

Anexo A: Tablas de la base de datos

En este anexo se presentan todas las tablas que forman parte de la base de datos con sus correspondientes atributos, claves primarias y claves foráneas. De cada atributo será indicado el tipo de datos.

Nombre del campo	Tipo de datos
Id	Número
Partido (FK)	Número
Id_Evento	Número
Cierre_Evento	Fecha/Hora
Descripcion	Texto
Prevision_Inicio_Evento	Fecha/Hora
Evento	Texto
Inicio_Evento	Fecha/Hora
Id_Apuesta	Número
Apuesta	Texto
Cuota	Número
Numero_apuestas	Número
Volumen_cruzado	Número

Nombre del campo	Tipo de datos
Ultima_apuesta	Fecha/Hora
Primera_apuesta	Fecha/Hora
Ganadora	Sí/No
En_juego	Texto {NI, PE, IP}

Tabla 73: Tabla Betfair.

Nombre del campo	Tipo de datos
Jugador (FK)	Número
Torneo (FK)	Número
Cabeza_de_serie	Texto

Tabla 74: Tabla Cabezas_de_serie

Nombre del campo	Tipo de datos
ID (FK)	Número
Jugador1 (FK)	Número
Jugador2 (FK)	Número
Torneo (FK)	Número
Ronda (FK)	Número
Primer_saque_metidos_1	Número
Primer_saque_realizados_1	Número
Aces_1	Número
Dobles_falta_1	Número
Errores_no_forzados_1	Número
Puntos_ganados_primer_saque_1	Número
Puntos_jugados_primer_saque_1	Número
Puntos_ganados_segundo_saque_1	Número
Puntos_jugados_segundo_saque_1	Número
Golpes_ganadores_1	Número

Nombre del campo	Tipo de datos
Puntos_de_break_convertidos_1	Número
Puntos_de_break_jugados_1	Número
Subidas_a_la_red_ganadas_1	Número
Subidas_a_la_red_1	Número
Total_puntos_ganados_1	Número
Velocidad_maxima_servicio_1	Número
Velocidad_media_primer_servicio_1	Número
Velocidad_media_segundo_servicio_1	Número
Primer_saque_metidos_2	Número
Primer_saque_realizados_2	Número
Aces_2	Número
Dobles_falta_2	Número
Errores_no_forzados_2	Número
Puntos_ganados_primer_saque_2	Número
Puntos_jugados_primer_saque_2	Número
Puntos_ganados_segundo_saque_2	Número
Puntos_jugados_segundo_saque_2	Número
Golpes_ganadores_2	Número
Puntos_de_break_convertidos_2	Número
Puntos_de_break_jugados_2	Número
Subidas_a_la_red_ganadas_2	Número
Subidas_a_la_red_2	Número
Total_puntos_ganados_2	Número
Velocidad_maxima_servicio_2	Número
Velocidad_media_primer_servicio_2	Número
Velocidad_media_segundo_servicio_2	Número
Puntos_restados_ganados_1	Número

Nombre del campo	Tipo de datos
Puntos_restados_1	Número
Puntos_restados_ganados_2	Número
Puntos_restados_2	Número

Tabla 75: Tabla Estadisticas_partidos.

Nombre del campo	Tipo de datos
ID	Número
Nombre	Texto
Fecha_nacimiento	Fecha/Hora
País	Texto
Ranking_actual_ATP	Número
Progresion	Número
Puntos	Número
Puntos_pista_dura	Número
Torneos_pista_dura	Número
Puntos_pista_tierra_batida	Número
Torneos_pista_tierra_batida	Número
Puntos_pista_hierba	Número
Torneos_pista_hierba	Número
Puntos_pista_moqueta	Número
Torneos_pista_moqueta	Número
Premios	Número
Puntos_carrera_de_campeones	Número
Ranking_actual_dobles_ATP	Número
Progresion_dobles	Número
Puntos_en_dobles	Número
Puntos_pista_dura_indoor	Número

Nombre del campo	Tipo de datos
Torneos_pista_dura_indoor	Número
ITF_ID	Número

Tabla 76: Tabla Jugadores.

Nombre del campo	Tipo de datos
ID	Número
Jugador (FK)	Número
Fecha	Fecha/Hora
Descripcion	Texto

Tabla 77: Tabla Lesiones.

Nombre del campo	Tipo de datos
ID	Número
Lenguaje	Número
Categoria	Texto
Nombre	Texto
Link	Texto

Tabla 78: Tabla Links.

Nombre del campo	Tipo de datos
ID	Número
Jugador1 (FK)	Número
Jugador2 (FK)	Número
Torneo (FK)	Número
Ronda (FK)	Número
Resultado	Texto
Fecha	Fecha/Hora

Tabla 79: Tabla Partidos.

Nombre del campo	Tipo de datos
ID	Número
Nombre_Pista	Texto

Tabla 80: Tabla Pistas.

Nombre del campo	Tipo de datos
ID	Número
Ganador	Número
Finalista	Número
Semifinalista	Número
Cuartofinalista	Número
Cuarta_ronda	Número
Tercera_ronda	Número
Segunda_ronda	Número
Primera_ronda	Número
Clasificado	Número
Tercera_ronda_clasificacion	Número
Segunda_ronda_clasificacion	Número
Primera_ronda_clasificacion	Número

Tabla 81: Tabla Puntuaciones.

Nombre del campo	Tipo de datos
Fecha	Fecha/Hora
Jugador (FK)	Número
Puntos	Número
Posicion	Número

Tabla 82: Tabla Ratings.

Nombre del campo	Tipo de datos
Jugador (FK)	Número
Año	Número
TMW_F	Número
TML_F	Número
TBW_F	Número
TBL_F	Número
MF_F	Número
Aces	Número
Dobles_falta	Número
Porcentaje_primer_servicio	Número
Porcentaje_puntos_ganados_primer_servicio	Número
Porcentaje_puntos_ganados_segundo_servicio	Número
SGW_F	Número
BPS_F	Número
Porcentaje_restos_ganados_primer_servicio	Número
Porcentaje_restos_ganados_segundo_servicio	Número
BPW_F	Número
RGW_F	Número
Premios	Número

Tabla 83: Tabla Resumen_datos.

Nombre del campo	Tipo de datos
ID	Número
Nombre	Texto

Tabla 84: Tabla Rondas.

Nombre del campo	Tipo de datos
ID	Número

Nombre del campo	Tipo de datos
Nombre	Texto
Tipo_pista (FK)	Número
Fecha	Fecha/Hora
Ranking	Número
Link	Número
Pais	Texto
Premios	Texto
Puntuacion (FK)	Número
URL	Texto
Latitud	Número
Longitud	Número
SITE_T	Texto
Puntuacion_carrera_campeones	Número
Puntuacion_lista_de_entradas	Número
SINGLES_T	Número
DOUBLES_T	Número
TIER_T	Texto
RESERVE_CHAR_T	Texto

Tabla 85: Tabla Torneos.

Nombre del campo	Tipo de datos
ID	Número
Link	Texto

Tabla 86: Tabla Urls.

Anexo B: Atributos del Experimento 1

A continuación, se muestra una tabla en la que se detallan los atributos que componen cada uno de los ejemplos del Experimento 1.

Atributo/s	Descripción
jugador1 / jugador2	Identifica a los jugadores que disputan el partido. Es el ID de la tabla `Jugadores` de cada uno de los jugadores que disputan el partido.
edadJ1 / edad J2	Edad de cada uno de los jugadores en el instante de la disputa del partido. Se obtuvo con la resta de la fecha del partido, obtenida del campo `Fecha` en la tabla `Partidos`, y de la fecha de nacimiento de cada uno de los jugadores, obtenida del campo `Fecha_nacimiento` de la tabla Jugadores.
puntosJ1 / puntosJ2	Puntos en la clasificación ATP de cada uno de los jugadores en la semana en la que se disputó el partido. Estos datos se obtienen consultando en la tabla `Ratings` los puntos de los jugadores en la fecha inmediatamente anterior a la fecha de disputa del partido.
posJ1 / posJ2	Posición en la clasificación ATP de cada uno de los jugadores en la semana en la que se disputó el partido. Estos datos se obtienen consultando en la tabla `Ratings` las posiciones de los jugadores en la fecha inmediatamente anterior a la fecha de disputa del partido.
puntosPasadoJ1 / puntosPasadoJ2	Puntos en la clasificación ATP de cada uno de los jugadores un año antes a la semana en la que se disputó el partido. Estos datos se obtienen consultando en la tabla `Ratings` los puntos de los jugadores en la fecha inmediatamente anterior a la fecha de disputa del partido menos un año.
posPasadoJ1 / posPasadoJ2	Posición en la clasificación ATP de cada uno de los jugadores un año antes a la semana en la que se disputó el partido. Estos datos se obtienen consultando en la tabla `Ratings` las posiciones de los jugadores en la fecha inmediatamente anterior a la fecha de disputa del partido menos un año.
progresionPuntosJ1 / progresionPuntosJ2	Progresión de los puntos de la clasificación ATP de los jugadores en el año previo a la disputa del partido. Se calcula restando los puntos de los jugadores en el momento del partido a los que tenían hace un año. Es decir `puntosJX – puntosPasadoJX`. Si la progresión ha sido positiva, es decir, en el momento del partido tienen más puntos que un año atrás, el valor del atributo será positivo.
progresionPosJ1 / progresionPosJ2	Progresión de la posición en la clasificación ATP de los jugadores en el año previo a la disputa del partido. Se calcula restando las posiciones de los jugadores un año antes de la disputa del partido

Atributo/s	Descripción
	a las que tenían en el momento del partido. Es decir `posPasadoJX – posJX`. Si la progresión ha sido positiva, es decir, en el momento del partido están en una posición más alta de la clasificación ATP que un año atrás, el valor del atributo será positivo.
TMWJ1 / TMWJ2 TMLJ1 / TMLJ2 TBWJ1 / TBWJ2 TBLJ1 / TBLJ2 MFJ1 / MFJ2 AcesJ1 / AcesJ2 DFaltaJ1 / DFaltaJ2 PPSJ1 / PPSJ2 PPGPSJ1 / PPGPSJ2 PPGSSJ1 / PPGSSJ2 SGWJ1 / SGWJ2 BPSJ1 / BPSJ2 PRGPSJ1 / PRGPSJ2 PRGSSJ1 / PRGSSJ2 BPWJ1 /BPWJ2 RGWJ1 /RGWJ2 PremiosJ1 / PremiosJ2	Todos estos atributos son los datos estadísticos de los partidos acumulados en los partidos disputados por cada uno de los jugadores en el año natural anterior a la fecha de disputa del partido. Son todos obtenidos de la tabla `Resumen_datos`.
CabezaSerieJ1 / CabezaSerieJ2	Estos atributos son de tipo nominal y sólo toman valores dentro del siguiente rango: `{0, 1, 10, 11, 11W, 12, 13, 14, 15, 15W, 16, 16W, 17, 18, 19, 1LL, 1q, 1WC, 2, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2LL, 2q, 2SE, 2WC, 3, 30, 30W, 31, 32, 33, 3q, 3WC, 4, 4LL, 4q, 4SE, 4WC, 5, 5LL, 5q, 5WC, 6, 6q, 6SE, 6WC, 7, 7WC, 8, 8WC, 9, 9WC, ALT, LL, PR, q, SE, WC}`. Si un jugador no es cabeza de serie de un torneo el atributo tomará el valor `0`. Estos atributos son obtenidos de la tabla `Cabezas_de_serie` a través del `ID` del torneo y del jugador.
RachaJ1 / RachaJ2	Indican el número de partidos seguidos que llevan los jugadores perdidos o ganados. Si el valor es positivo, significa que la racha es de partidos ganados y si es negativo significa que es de partidos perdidos. Este atributo ha sido calculado consultando los partidos, inmediatamente anteriores en fecha al del ejemplo, de cada uno de los jugadores.
ultimos5GanadosJ1 / ultimos5GanadosJ2	Indica de los últimos cinco partidos disputados inmediatamente antes al del ejemplo cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos5PerdidosJ1 / ultimos5PerdidosJ2	Indica de los últimos cinco partidos disputados inmediatamente antes al del ejemplo cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos10GanadosJ1 /	Indica de los últimos diez partidos disputados inmediatamente

Atributo/s	Descripción
ultimos10GanadosJ2	antes al del ejemplo cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos10PerdidosJ1 / ultimos10PerdidosJ2	Indica de los últimos diez partidos disputados inmediatamente antes al del ejemplo cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos20GanadosJ1 / ultimos20GanadosJ2	Indica de los últimos veinte partidos disputados inmediatamente antes al del ejemplo cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos20PerdidosJ1 / ultimos20PerdidosJ2	Indica de los últimos veinte partidos disputados inmediatamente antes al del ejemplo cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos30GanadosJ1 / ultimos30GanadosJ2	Indica de los últimos treinta partidos disputados inmediatamente antes al del ejemplo cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos30PerdidosJ1 / ultimos30PerdidosJ2	Indica de los últimos treinta partidos disputados inmediatamente antes al del ejemplo cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos50GanadosJ1 / ultimos50GanadosJ2	Indica de los últimos cincuenta partidos disputados inmediatamente antes al del ejemplo cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos50PerdidosJ1 / ultimos50PerdidosJ2	Indica de los últimos cincuenta partidos disputados inmediatamente antes al del ejemplo cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ganadosAnioJ1 / ganadosAnioJ2	Indica de todos los partidos disputados por cada uno de los jugadores en ese año natural, hasta el momento del partido del ejemplo, cuántos han ganado. Estos partidos son consultados en la tabla `Partidos`.
perdidosAnioJ1 / perdidosAnioJ2	Indica de todos los partidos disputados por cada uno de los jugadores en ese año natural, hasta el momento del partido del ejemplo, cuántos han perdido. Estos partidos son consultados en la tabla `Partidos`.
ganadosAnioAnteriorJ1 / ganadosAnioAnteriorJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el año natural anterior al partido del ejemplo, cuántos han ganado. Estos partidos son consultados en la tabla `Partidos`.
perdidosAnioAnteriorJ1 / perdidosAnioAnteriorJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el año natural anterior al partido del ejemplo, cuántos han perdido. Estos partidos son consultados en la tabla

Atributo/s	Descripción
	Partidos.
ganadosJ1 / ganadosJ2	Indica de todos los partidos disputados por cada uno de los jugadores en toda su carrera como profesional, hasta el momento del partido del ejemplo, cuántos han ganado. Estos partidos son consultados en la tabla Partidos.
perdidosJ1 / perdidosJ2	Indica de todos los partidos disputados por cada uno de los jugadores en toda su carrera como profesional, hasta el momento del partido del ejemplo, cuántos han perdido. Estos partidos son consultados en la tabla Partidos.
ganadosMesJ1 / ganadosMesJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el mismo mes de la disputa del partido en toda su carrera como profesional, hasta el momento del partido del ejemplo, cuántos han ganado. Estos partidos son consultados en la tabla Partidos.
perdidosMesJ1 / perdidosMesJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el mismo mes de la disputa del partido en toda su carrera como profesional, hasta el momento del partido del ejemplo, cuántos han perdido. Estos partidos son consultados en la tabla Partidos.
ganadosPeriodoJ1 / ganadosPeriodoJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el mismo mes y en el mes siguiente y anterior de la disputa del partido en toda su carrera como profesional, hasta el momento del partido del ejemplo, cuántos han ganado. Estos partidos son consultados en la tabla Partidos.
perdidosPeriodoJ1 / perdidosPeriodoJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el mismo mes y en el mes siguiente y anterior de la disputa del partido en toda su carrera como profesional, hasta el momento del partido del ejemplo, cuántos han perdido. Estos partidos son consultados en la tabla Partidos.
ganadosRondaJ1 / ganadosRondaJ2	Indica de todos los partidos disputados por cada uno de los jugadores en la misma ronda de cualquier torneo, hasta el momento del partido del ejemplo, cuántos han ganado. Estos partidos son consultados en la tabla Partidos.
perdidosRondaJ1 / perdidosRondaJ2	Indica de todos los partidos disputados por cada uno de los jugadores en la misma ronda de cualquier torneo, hasta el momento del partido del ejemplo, cuántos han perdido. Estos partidos son consultados en la tabla Partidos.
RachaSuperficieJ1 / RachaSuperficieJ2	Indican el número de partidos seguidos que llevan los jugadores perdidos o ganados en la misma superficie en la que se disputa el partido. Si el valor es positivo significa que la racha es de partidos ganados y si es negativo significa que es de partidos perdidos. Este atributo ha sido calculado consultando los partidos,

Atributo/s	Descripción
	inmediatamente anteriores en fecha al del ejemplo, de cada uno de los jugadores.
ultimos5SupGanadosJ1 / ultimos5SupGanadosJ2	Indica de los últimos cinco partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos5SupPerdidosJ1 / ultimos5SupPerdidosJ2	Indica de los últimos cinco partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos10SupGanadosJ1 / ultimos10SupGanadosJ2	Indica de los últimos diez partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos10SupPerdidosJ1 / ultimos10SupPerdidosJ2	Indica de los últimos diez partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos20SupGanadosJ1 / ultimos20SupGanadosJ2	Indica de los últimos veinte partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos20SupPerdidosJ1 / ultimos20SupPerdidosJ2	Indica de los últimos veinte partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos30SupGanadosJ1 / ultimos30SupGanadosJ2	Indica de los últimos treinta partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos30SupPerdidosJ1 / ultimos30SupPerdidosJ2	Indica de los últimos treinta partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han perdido cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos50SupGanadosJ1 / ultimos50SupGanadosJ2	Indica de los últimos cincuenta partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han ganado cada uno de los jugadores. Estos partidos son consultados en la tabla `Partidos`.
ultimos50SupPerdidosJ1 / ultimos50SupPerdidosJ2	Indica de los últimos cincuenta partidos disputados inmediatamente antes al del ejemplo y en la misma superficie en la que se disputa el partido cuántos han perdido cada uno de los

Atributo/s	Descripción
	jugadores. Estos partidos son consultados en la tabla `Partidos`.
ganadosAnioSuperficieJ1 / ganadosAnioSuperficieJ2	Indica de todos los partidos disputados por cada uno de los jugadores en ese año natural, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han ganado. Estos partidos son consultados en la tabla `Partidos`.
perdidosAnioSuperficieJ1 / perdidosAnioSuperficieJ2	Indica de todos los partidos disputados por cada uno de los jugadores en ese año natural, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han perdido. Estos partidos son consultados en la tabla `Partidos`.
ganadosAnioAntSupJ1 / ganadosAnioAntSupJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el año natural anterior al partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han ganado. Estos partidos son consultados en la tabla `Partidos`.
perdidosAnioAntSupJ1 / perdidosAnioAntSupJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el año natural anterior al partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han perdido. Estos partidos son consultados en la tabla `Partidos`.
ganadosSuperficieJ1 / ganadosSuperficieJ2	Indica de todos los partidos disputados por cada uno de los jugadores en toda su carrera como profesional, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han ganado. Estos partidos son consultados en la tabla `Partidos`.
perdidosSuperficieJ1 / perdidosSuperficieJ2	Indica de todos los partidos disputados por cada uno de los jugadores en toda su carrera como profesional, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han perdido. Estos partidos son consultados en la tabla `Partidos`.
ganadosMesSuperficieJ1 / ganadosMesSuperficieJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el mismo mes de la disputa del partido en toda su carrera como profesional, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han ganado. Estos partidos son consultados en la tabla `Partidos`.
perdidosMesSuperficieJ1 / perdidosMesSuperficieJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el mismo mes de la disputa del partido en toda su carrera como profesional, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han perdido. Estos partidos son consultados en la tabla `Partidos`.
ganadosPeriodoSupJ1 /	Indica de todos los partidos disputados por cada uno de los

Atributo/s	Descripción
ganadosPeriodoSupJ2	jugadores en el mismo mes y en el mes siguiente y anterior de la disputa del partido en toda su carrera como profesional, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han ganado. Estos partidos son consultados en la tabla `Partidos`.
perdidosPeriodoSupJ1 / perdidosPeriodoSupJ2	Indica de todos los partidos disputados por cada uno de los jugadores en el mismo mes y en el mes siguiente y anterior de la disputa del partido en toda su carrera como profesional, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han perdido. Estos partidos son consultados en la tabla `Partidos`.
ganadosRondaSupJ1 / ganadosRondaSupJ2	Indica de todos los partidos disputados por cada uno de los jugadores en la misma ronda de cualquier torneo, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han ganado. Estos partidos son consultados en la tabla `Partidos`.
perdidosRondaSupJ1 / perdidosRondaSupJ2	Indica de todos los partidos disputados por cada uno de los jugadores en la misma ronda de cualquier torneo, hasta el momento del partido del ejemplo y en la misma superficie en la que se disputa el partido, cuántos han perdido. Estos partidos son consultados en la tabla `Partidos`.
caraAcaraGanados	Refleja el número de partidos ganados por el jugador 1 en enfrentamientos directos contra el jugador 2 antes del partido del ejemplo. Estos partidos son consultados en la tabla `Partidos`.
caraAcaraPerdidos	Refleja el número de partidos perdidos por el jugador 1 en enfrentamientos directos contra el jugador 2 antes del partido del ejemplo. Estos partidos son consultados en la tabla `Partidos`.
caraAcaraGanadosSup	Refleja el número de partidos ganados por el jugador 1 en enfrentamientos directos contra el jugador 2 antes del partido del ejemplo y en la misma superficie en la que se disputa el partido. Estos partidos son consultados en la tabla `Partidos`.
caraAcaraPerdidosSup	Refleja el número de partidos perdidos por el jugador 1 en enfrentamientos directos contra el jugador 2 antes del partido del ejemplo y en la misma superficie en la que se disputa el partido. Estos partidos son consultados en la tabla `Partidos`.
Torneo	Identifica al torneo al que pertenece el partido. Es el `ID` de la tabla Torneos.
Ronda	Identifica a la ronda en la que se disputa el partido. Es el `ID` de la tabla `Rondas` y su valor es nominal y pertenece al conjunto `{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17}`.
pista	Identifica a la pista/superficie en la que se disputa el partido. Es el

Atributo/s	Descripción
	`ID` de la tabla `Pistas` y su valor es nominal y pertenece al conjunto `{1,2,3,4,5,6}`.
rankingTorneo	Identifica el ranking al que pertenece el torneo en el que se disputa el partido.
premiosTorneo	Es el único atributo de tipo cadena de caracteres. Es un campo de la tabla torneos y es el montante total en premios que se reparte en el torneo. Está en cadena de caracteres porque en él se almacena la moneda en la que son los premios ($, €…) y si es un premio en miles, millones… El atributo ha sido eliminado para la experimentación porque causaba problemas al trabajar con algoritmos que no soportaban tipos de datos de cadena de caracteres.
ganador	Es la clase del problema. Sólo toma dos valores 1 o 2. Si toma el valor 1 es que el ganador del partido ha sido el Jugador1 y si toma el valor 2 el ganador ha sido el jugador 2.

Tabla 87: Atributos del archivo Weka para el Experimento 1.

Anexo C: Atributos del Experimento 2

A continuación, se muestra una tabla en la que se detallan los atributos que componen cada uno de los ejemplos del Experimento 2.

Atributo/s	Descripción
Cuota	Cuota a la que se había apostado en esa línea, es decir, lo que se paga por esa apuesta. Es de tipo real.
Numero_Apuestas	Número de apuestas que han sido igualadas a esa cuota. Suele estar directamente relacionada con el número de usuarios que han apostado aunque siempre se puede dar el caso de que un usuario en vez de realizar una única apuesta de 10 € realice 10 apuestas de 1€. Es de tipo numérico.
Volumen	Cantidad total de dinero apostado a esa apuesta. Es de tipo real.
Ganada	Este atributo será la clase del experimento y sólo cuenta con dos posibles valores `{T, F}`. Si es `T` es que la apuesta resultó ganadora y en caso contrario la apuesta habría sido perdedora.

Tabla 88: Atributos del archivo Weka para el Experimento 2.

Anexo D: Atributos del Experimento 3

A continuación, se muestra una tabla en la que se detallan los atributos que componen cada uno de los ejemplos del Experimento 3.

Atributo/s	Descripción
Cuota	Cuota a la que se había apostado en esa línea, es decir, lo que se paga por esa apuesta. Es de tipo real.
Numero_Apuestas	Número de apuestas que han sido igualadas a esa cuota. Suele estar directamente relacionada con el número de usuarios que han apostado aunque siempre se puede dar el caso de que un usuario en vez de realizar una única apuesta de 10 € realice 10 apuestas de 1€. Es de tipo numérico.
Volumen	Cantidad total de dinero apostado a esa apuesta. Es de tipo real.
EnJuego	Indica si la apuesta ha sido realizada antes del comienzo del partido porque en vivo no hubiera sido posible (`NI`) o porque fue decisión del apostante el no apostar en vivo (`PE`). Por tanto, sólo puede tener dos posibles valores `{P, N}`.
Ganada	Este atributo será la clase del experimento y sólo cuenta con dos posibles valores `{T, F}`. Si es `T` es que la apuesta resultó ganadora y en caso contrario la apuesta habría sido perdedora.

Tabla 89: Atributos del archivo Weka para el Experimento 3.

Anexo E: Atributos del Experimento 5

A continuación, se muestra una tabla en la que se detallan los atributos que componen cada uno de los ejemplos del Experimento 5.

Atributo/s	Descripción
Todos los atributos del experimento 1 (*Anexo B: Atributos del Experimento 1*) a excepción del atributo `premiosTorneo` forman parte de este experimento.	
CuotaMaxJ1 / CuotaMaxJ2	Máximo valor de la cuota al que se ha realizado alguna apuesta antes del inicio de los partidos a favor de cada uno de los jugadores.
CuotaMinJ1 /	Mínimo valor de la cuota al que se ha realizado alguna apuesta

Atributo/s	Descripción
CuotaMinJ2	antes del inicio de los partidos a favor de cada uno de los jugadores.
CuotaMediaJ1 / CuotaMediaJ2	Valor medio de la cuota a favor de cada uno de los jugadores antes del inicio de los partidos. Para calcular este valor además del valor de las cuotas de las apuesta debe ser tenido en cuenta el volumen apostada a cada cuota.
ultimaCuotaJ1 / ultimaCuotaJ2	Valor de la cuota a la que se realizó la última apuesta antes del inicio del partido.

Tabla 90: Atributos del archivo Weka para el Experimento 5.

Anexo F: Atributos del Experimento 6

A continuación, se muestra una tabla en la que se detallan los atributos que componen cada uno de los ejemplos del experimento 6.

Atributo/s	Descripción
CuotaMaxJ1 / CuotaMaxJ2	Máximo valor de la cuota al que se ha realizado alguna apuesta antes del inicio de los partidos a favor de cada uno de los jugadores.
CuotaMinJ1 / CuotaMinJ2	Mínimo valor de la cuota al que se ha realizado alguna apuesta antes del inicio de los partidos a favor de cada uno de los jugadores.
CuotaMediaJ1 / CuotaMediaJ2	Valor medio de la cuota a favor de cada uno de los jugadores antes del inicio de los partidos. Para calcular este valor además del valor de las cuotas de las apuesta debe ser tenido en cuenta el volumen apostada a cada cuota.
ultimaCuotaJ1 / ultimaCuotaJ2	Valor de la cuota a la que se realizó la última apuesta antes del inicio del partido.
idPartido	Identifica unívocamente el partido en la base de datos en la tabla Partidos.
jugador1 / jugador2	Identifica a los jugadores que disputan el partido. Es el ID de la tabla Jugadores de cada uno de los jugadores que disputan el partido.
edad	Edad del jugador 1 – Edad del jugador 2.

Atributo/s	Descripción
puntos	Puntos del jugador 1 – Puntos del jugador 2.
pos	Posición del jugador 2 – Posición del jugador 1.
puntosPasado	Puntos del jugador 1 un año antes – Puntos del jugador 2 un año antes.
posPasado	Posición del jugador 2 un año antes – Posición del jugador 1 un año antes.
progresionPuntos	Progresión de los puntos del jugador 1 – Progresión de los puntos del jugador 2.
progresionPosJ1 / progresionPosJ2	Progresión de la posición de la posición del jugador 1 – Progresión de la posición del jugador 2
TMW TML TBW TBL MF Aces DFalta PPS PPGPS PPGSS SGW BPS PRGPS PRGSS BPW RGW Premios	Todos estos atributos son el resultado de restas los datos estadísticos de los partidos acumulados en los partidos disputados por el jugador 1 a los mismos datos del jugador 2.
CabezaSerieJ1 / CabezaSerieJ2	Estos atributos son de tipo nominal y sólo toman valores dentro del siguiente rango: {0, 1, 10, 11, 11W, 12, 13, 14, 15, 15W, 16, 16W, 17, 18, 19, 1LL, 1q, 1WC, 2, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2LL, 2q, 2SE, 2WC, 3, 30, 30W, 31, 32, 33, 3q, 3WC, 4, 4LL, 4q, 4SE, 4WC, 5, 5LL, 5q, 5WC, 6, 6q, 6SE, 6WC, 7, 7WC, 8, 8WC, 9, 9WC, ALT, LL, PR, q, SE, WC}. Si un jugador no es cabeza de serie de un torneo el atributo tomará el valor 0. Estos atributos son obtenidos de la tabla Cabezas_de_serie a través del ID del torneo y del jugador.
Racha	Racha del jugador 1 – Racha del jugador 2.
ultimos5Ganados	Últimos 5 partidos ganados del jugador 1 - Últimos 5 partidos ganados del jugador 2.
ultimos10Ganados	Últimos 10 partidos ganados del jugador 1 - Últimos 10 partidos ganados del jugador 2

Atributo/s	Descripción
ultimos20Ganados	Últimos 20 partidos ganados del jugador 1 - Últimos 20 partidos ganados del jugador 2.
ultimos30Ganados	Últimos 30 partidos ganados del jugador 1 - Últimos 30 partidos ganados del jugador 2.
ultimos50Ganados	Últimos 50 partidos ganados del jugador 1 - Últimos 50 partidos ganados del jugador 2.
ganadosAnio	Partidos ganados en el año en curso por el jugador 1 - Partidos ganados en el año en curso por el jugador 2.
perdidosAnio	Partidos perdidos en el año en curso por el jugador 1 - Partidos perdidos en el año en curso por el jugador 2.
ganadosAnioAnterior	Partidos ganados en el año natural anterior por el jugador 1 - Partidos ganados en el año natural anterior por el jugador 2.
perdidosAnioAnterior	Partidos perdidos en el año natural anterior por el jugador 1 - Partidos perdidos en el año natural anterior por el jugador 2.
ganados	Total de partidos ganados por el jugador 1 en su carrera - Total de partidos ganados por el jugador 2 en su carrera
perdidos	Total de partidos perdidos por el jugador 1 en su carrera - Total de partidos perdidos por el jugador 2 en su carrera
ganadosMes	Total de partidos ganados por el jugador 1 en el mes en curso - Total de partidos ganados por el jugador 2 en el mes en curso.
perdidosMes	Total de partidos perdidos por el jugador 1 en el mes en curso - Total de partidos perdidos por el jugador 2 en el mes en curso.
ganadosPeriodo	Total de partidos ganados en el mes en curso y en los meses posterior y anterior por el jugador 1 - Total de partidos ganados en el mes en curso y en los meses posterior y anterior por el jugador 2.
perdidosPeriodo	Total de partidos perdidos en el mes en curso y en los meses posterior y anterior por el jugador 1 - Total de partidos perdidos en el mes en curso y en los meses posterior y anterior por el jugador 2.
ganadosRonda	Total de partidos ganados en la ronda por el jugador 1 - Total de partidos ganados en la ronda por el jugador 2.
perdidosRonda	Total de partidos perdidos en la ronda por el jugador 1 - Total de partidos perdidos en la ronda por el jugador 2.
RachaSuperficie	Racha en la superficie del jugador 1 – Racha en la superficie del jugador 2.

Atributo/s	Descripción
ultimos5SupGanados	Últimos 5 partidos ganados en la superficie del jugador 1 - Últimos 5 partidos ganados en la superficie del jugador 2.
ultimos10SupGanados	Últimos 10 partidos ganados en la superficie del jugador 1 - Últimos 10 partidos ganados en la superficie del jugador 2
ultimos20SupGanados	Últimos 20 partidos ganados en la superficie del jugador 1 - Últimos 20 partidos ganados en la superficie del jugador 2.
ultimos30SupGanados	Últimos 30 partidos ganados en la superficie del jugador 1 - Últimos 30 partidos ganados en la superficie del jugador 2.
ultimos50SupGanados	Últimos 50 partidos ganados en la superficie del jugador 1 - Últimos 50 partidos ganados en la superficie del jugador 2.
ganadosAnioSuperficie	Partidos ganados en la superficie en el año en curso por el jugador 1 - Partidos ganados en la superficie en el año en curso por el jugador 2.
perdidosAnioSuperficie	Partidos perdidos en la superficie en el año en curso por el jugador 1 - Partidos perdidos en la superficie en el año en curso por el jugador 2.
ganadosAnioAntSup	Partidos ganados en la superficie en el año natural anterior por el jugador 1 - Partidos ganados en la superficie en el año natural anterior por el jugador 2.
perdidosAnioAntSup	Partidos perdidos en la superficie en el año natural anterior por el jugador 1 - Partidos perdidos en la superficie en el año natural anterior por el jugador 2.
ganadosSuperficie	Total de partidos ganados en la superficie por el jugador 1 en su carrera - Total de partidos ganados en la superficie por el jugador 2 en su carrera
perdidosSuperficie	Total de partidos perdidos en la superficie por el jugador 1 en su carrera - Total de partidos perdidos en la superficie por el jugador 2 en su carrera
ganadosMesSuperficie	Total de partidos ganados en la superficie por el jugador 1 en el mes en curso - Total de partidos ganados en la superficie por el jugador 2 en el mes en curso.
perdidosMesSuperficie	Total de partidos perdidos en la superficie por el jugador 1 en el mes en curso - Total de partidos perdidos en la superficie por el jugador 2 en el mes en curso.
ganadosPeriodoSup	Total de partidos ganados en la superficie en el mes en curso y en los meses posterior y anterior por el jugador 1 - Total de partidos ganados en la superficie en el mes en curso y en los meses posterior y anterior por el jugador 2.

Atributo/s	Descripción
perdidosPeriodoSup	Total de partidos perdidos en la superficie en el mes en curso y en los meses posterior y anterior por el jugador 1 - Total de partidos perdidos en la superficie en el mes en curso y en los meses posterior y anterior por el jugador 2.
ganadosRondaSup	Total de partidos ganados en la superficie en la ronda por el jugador 1 - Total de partidos ganados en la superficie en la ronda por el jugador 2.
perdidosRondaSup	Total de partidos perdidos en la superficie en la ronda por el jugador 1 - Total de partidos perdidos en la superficie en la ronda por el jugador 2.
caraAcara	Partidos ganados por el jugador 1 al jugador 2 - Partidos ganados por el jugador 2 al jugador 1.
caraAcaraSuperficie	Partidos ganados en la superficie por el jugador 1 al jugador 2 - Partidos ganados en la superficie por el jugador 2 al jugador 1.
Ronda	Identifica a la ronda en la que se disputa el partido. Es el ID de la tabla Rondas y su valor es nominal y pertenece al conjunto `{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17}`.
pista	Identifica a la pista/superficie en la que se disputa el partido. Es el `ID` de la tabla `Pistas` y su valor es nominal y pertenece al conjunto `{1,2,3,4,5,6}`.
rankingTorneo	Identifica el ranking al que pertenece el torneo en el que se disputa el partido.
ganador	Es la clase del problema. Sólo toma dos valores 1 o 2. Si toma el valor 1 es que el ganador del partido ha sido el Jugador1 y si toma el valor 2 el ganador ha sido el jugador 2.

Tabla 91: Atributos del archivo weka para el Experimento 6.

Anexo G: Resultados de las evaluaciones con el criterio de Kelly

Multiplicador	Balance	Balance sin picos	Rentabilidad	Invertido (Nº de partidos)	Predic. errónea
0,01	76,64	35,08	6,52%	1175	410
0,02	154,77	71,79	13,17%	1175	410

Multiplicador	Balance	Balance sin picos	Rentabilidad	Invertido (Nº de partidos)	Predic. errónea
0,03	234,37	110,12	19,95%	1175	410
0,04	315,45	150,11	26,85%	1175	410
0,05	397,93	191,72	33,87%	1175	410
0,06	481,89	235,06	41,01%	1175	410
0,07	567,24	280,06	48,28%	1175	410
0,08	654	326,77	55,66%	1175	410
0,09	742,16	375,2	63,16%	1175	410
0,1	831,64	425,32	70,78%	1175	410
0,11	922,49	477,18	78,51%	1175	410
0,12	1014,65	530,76	86,35%	1175	410
0,13	1108,14	586,1	94,31%	1175	410
0,14	1202,85	643,13	102,37%	1175	410
0,15	1298,86	701,93	110,54%	1175	410
0,16	1396,04	762,42	118,81%	1175	410
0,17	1494,42	824,63	127,18%	1175	410
0,18	1593,96	888,56	135,66%	1175	410
0,19	1694,62	954,19	144,22%	1175	410
0,2	1796,37	1021,52	152,88%	1175	410
0,21	1899,15	1090,5	161,63%	1175	410
0,22	2002,98	1161,17	170,47%	1175	410
0,23	2107,76	1233,46	179,38%	1175	410
0,24	2213,49	1307,38	188,38%	1175	410
0,25	2320,12	1382,9	197,46%	1175	410
0,26	2427,58	1459,99	206,60%	1175	410
0,27	2535,86	1538,64	215,82%	1175	410
0,28	2644,9	1618,81	225,10%	1175	410

Multiplicador	Balance	Balance sin picos	Rentabilidad	Invertido (Nº de partidos)	Predic. errónea
0,29	2754,67	1700,49	234,44%	1175	410
0,3	2865,11	1783,64	243,84%	1175	410
0,31	2976,14	1868,18	253,29%	1175	410
0,32	3087,73	1954,12	262,79%	1175	410
0,33	3199,86	2041,42	272,33%	1175	410
0,34	3312,42	2130,03	281,91%	1175	410
0,35	3425,38	2219,91	291,52%	1175	410
0,36	3538,7	2311,02	301,17%	1175	410
0,37	3652,28	2403,3	310,83%	1175	410
0,38	3766,12	2496,73	320,52%	1175	410
0,39	3880,11	2591,24	330,22%	1175	410
0,4	3994,18	2686,76	339,93%	1175	410
0,41	4108,33	2783,28	349,65%	1175	410
0,42	4222,43	2880,72	359,36%	1175	410
0,43	4336,42	2979	369,06%	1175	410
0,44	4450,3	3078,12	378,75%	1175	410
0,45	4563,89	3177,94	388,42%	1175	410
0,46	4677,24	3278,48	398,06%	1175	410
0,47	4790,23	3379,65	407,68%	1175	410
0,48	4902,75	3481,33	417,26%	1175	410
0,49	5014,77	3583,51	426,79%	1175	410
0,5	5126,2	3686,09	436,27%	1175	410
0,51	5236,99	3789,04	445,70%	1175	410
0,52	5347,07	3892,26	455,07%	1175	410
0,53	5456,33	3995,68	464,37%	1175	410
0,54	5564,68	4073,59	473,59%	1175	410

Multiplicador	Balance	Balance sin picos	Rentabilidad	Invertido (Nº de partidos)	Predic. errónea
0,55	5672,13	4139,55	482,73%	1175	410
0,56	5778,5	4204,57	491,79%	1175	410
0,57	5883,76	4268,7	500,75%	1175	410
0,58	5987,85	4330,24	509,60%	1175	410
0,59	6090,67	4385,84	518,35%	1175	410
0,6	6192,13	4439,87	526,99%	1175	410
0,61	6292,16	4492,29	535,50%	1175	410
0,62	6390,68	4543,03	543,89%	1175	410
0,63	6487,62	4592,06	552,14%	1175	410
0,64	6582,89	4639,3	560,25%	1175	410
0,65	6676,4	4684,7	568,20%	1175	410
0,66	6768,1	4728,23	576,01%	1175	410
0,67	6857,86	4769,8	583,65%	1175	410
0,68	6945,63	4809,37	591,12%	1175	410
0,69	7031,34	4846,89	598,41%	1175	410
0,7	7114,89	4882,32	605,52%	1175	410
0,71	7196,19	4915,57	612,44%	1175	410
0,72	7275,18	4946,89	619,16%	1175	410
0,73	7351,77	4976,02	625,68%	1175	410
0,74	7425,89	5002,81	631,99%	1175	410
0,75	7497,46	5027,22	638,08%	1175	410
0,76	7566,37	5049,16	643,95%	1175	410
0,77	7632,61	5068,65	649,58%	1175	410
0,78	7696,04	5085,58	654,98%	1175	410
0,79	7756,61	5099,92	660,14%	1175	410
0,8	7814,22	5111,62	665,04%	1175	410

Multiplicador	Balance	Balance sin picos	Rentabilidad	Invertido (Nº de partidos)	Predic. errónea
0,81	7868,81	5120,63	669,69%	1175	410
0,82	7920,32	5126,94	674,07%	1175	410
0,83	7968,67	5130,48	678,18%	1175	410
0,84	8013,81	5131,22	682,03%	1175	410
0,85	8055,62	5129,1	685,58%	1175	410
0,86	8094,04	5124,07	688,85%	1175	410
0,87	8129,04	5116,13	691,83%	1175	410
0,88	8160,51	5105,2	694,51%	1175	410
0,89	8188,4	5091,26	696,89%	1175	410
0,9	8212,67	5074,3	698,95%	1175	410
0,91	8233,21	5054,25	700,70%	1175	410
0,92	8250,02	5031,11	702,13%	1175	410
0,93	8262,96	5004,8	703,23%	1175	410
0,94	8272,05	4975,34	704,00%	1175	410
0,95	8277,19	4942,68	704,44%	1175	410
0,96	8278,36	4906,8	704,54%	1175	410
0,97	8275,47	4867,68	704,30%	1175	410
0,98	8268,46	4825,26	703,70%	1175	410
0,99	8257,33	4779,56	702,75%	1175	410
1	8242,02	4730,57	701,45%	1175	410

Tabla 92: Resultados de la evaluación del Experimento 7 A utilizando como sistema de apuestas el criterio de Kelly.

Multiplicador	Balance Cuota media	Balance última cuota	Balance cuota máxima	Balance cuota mínima
0,01	0,4	85,56	4856,96	-187,24
0,02	-5,39	169,52	9910,35	-373,75
0,03	-17,65	251,14	15151,29	-559,58

Multiplicador	Balance Cuota media	Balance última cuota	Balance cuota máxima	Balance cuota mínima
0,04	-36,81	329,93	20570,42	-744,87
0,05	-63,11	405,06	26157,88	-929,74
0,06	-96,94	475,96	31903,28	-1114,24
0,07	-138,55	541,9	37795,7	-1298,43
0,08	-188,19	602,14	43823,76	-1482,39
0,09	-246,07	656,03	49975,4	-1666,18
0,1	-312,38	702,86	56238,06	-1849,87
0,11	-387,29	741,94	62598,49	-2033,41
0,12	-470,93	772,59	69042,81	-2216,95
0,13	-563,3	794,23	75556,4	-2400,45
0,14	-664,53	806,25	82123,94	-2583,9
0,15	-774,58	808,03	88729,35	-2767,34
0,16	-893,4	799,11	95355,69	-2950,76
0,17	-1020,92	778,95	101985,4	-3134,15
0,18	-1157,02	747,18	108599,86	-3317,51
0,19	-1301,58	703,34	115179,94	-3500,68
0,2	-1454,38	647,24	121705,54	-3683,8
0,21	-1615,24	578,52	128155,97	-3866,75
0,22	-1783,88	497,08	134509,82	-4049,46
0,23	-1960	402,72	140745,08	-4231,87
0,24	-2143,31	295,45	146839,3	-4413,95
0,25	-2333,5	175,23	152769,61	-4595,62
0,26	-2530,07	42,18	158512,94	-4776,84
0,27	-2732,75	-103,56	164046,03	-4957,44
0,28	-2941,11	-261,79	169345,76	-5137,43
0,29	-3154,7	-432,2	174389,22	-5316,72

Multiplicador	Balance Cuota media	Balance última cuota	Balance cuota máxima	Balance cuota mínima
0,3	-3373,07	-614,44	179153,92	-5495,18
0,31	-3595,78	-808,16	183617,99	-5672,72
0,32	-3822,37	-1012,91	187760,38	-5849,29
0,33	-4052,36	-1228,12	191561,14	-6024,77
0,34	-4285,23	-1453,29	195001,43	-6199,06
0,35	-4520,56	-1687,78	198064	-6372,08
0,36	-4757,84	-1930,99	200733,13	-6543,77
0,37	-4996,6	-2182,26	202995,08	-6713,96
0,38	-5236,39	-2440,83	204837,95	-6882,65
0,39	-5476,7	-2706,03	206252,11	-7049,65
0,4	-5717,08	-2977,13	207230,25	-7214,92
0,41	-5957,17	-3253,34	207767,43	-7378,39
0,42	-6196,43	-3533,95	207861,21	-7539,94
0,43	-6434,54	-3818,15	207511,8	-7699,51
0,44	-6671,02	-4105,2	206721,97	-7857
0,45	-6905,58	-4394,39	205497,12	-8012,35
0,46	-7137,78	-4684,91	203845,18	-8165,47
0,47	-7367,33	-4976,09	201776,69	-8316,33
0,48	-7593,93	-5267,19	199304,58	-8464,78
0,49	-7817,21	-5557,59	196444,17	-8610,84
0,5	-8036,98	-5846,6	193212,85	-8754,43
0,51	-8252,9	-6133,6	189630,17	-8895,47
0,52	-8464,82	-6417,98	185717,38	-9033,95
0,53	-8672,48	-6699,21	181497,44	-9169,76
0,54	-8875,74	-6976,77	176994,64	-9302,95
0,55	-9074,39	-7250,15	172234,47	-9433,39

Multiplicador	Balance Cuota media	Balance última cuota	Balance cuota máxima	Balance cuota mínima
0,56	-9268,33	-7518,95	167243,33	-9561,15
0,57	-9457,41	-7782,71	162048,21	-9686,09
0,58	-9641,52	-8041,11	156676,64	-9808,27
0,59	-9820,62	-8293,81	151156,11	-9927,67
0,6	-9994,63	-8540,48	145514,27	-10044,23
0,61	-10163,51	-8780,92	139778,14	-10157,97
0,62	-10327,22	-9014,9	133974,45	-10268,88
0,63	-10485,77	-9242,22	128129,07	-10376,98
0,64	-10639,14	-9462,76	122266,81	-10482,25
0,65	-10787,4	-9676,42	116411,62	-10584,67
0,66	-10930,52	-9883,09	110585,9	-10684,3
0,67	-11068,58	-10082,72	104810,85	-10781,17
0,68	-11201,63	-10275,36	99106,07	-10875,23
0,69	-11329,75	-10460,94	93489,6	-10966,53
0,7	-11453,02	-10639,52	87977,92	-11055,14
0,71	-11571,49	-10811,19	82585,68	-11141,07
0,72	-11685,27	-10975,96	77325,96	-11224,31
0,73	-11794,5	-11133,97	72210,15	-11304,91
0,74	-11899,21	-11285,29	67247,97	-11382,97
0,75	-11999,57	-11430,11	62447,55	-11458,45
0,76	-12095,7	-11568,55	57815,44	-11531,44
0,77	-12187,62	-11700,71	53356,74	-11601,94
0,78	-12275,58	-11826,78	49075,09	-11670,03
0,79	-12359,61	-11946,94	44972,98	-11735,76
0,8	-12439,87	-12061,34	41051,53	-11799,17
0,81	-12516,47	-12170,19	37310,82	-11860,32

Multiplicador	Balance Cuota media	Balance última cuota	Balance cuota máxima	Balance cuota mínima
0,82	-12589,53	-12273,66	33749,97	-11919,24
0,83	-12659,21	-12371,92	30367,14	-11975,99
0,84	-12725,57	-12465,16	27159,76	-12030,64
0,85	-12788,79	-12553,58	24124,53	-12083,24
0,86	-12848,96	-12637,38	21257,53	-12133,83
0,87	-12906,21	-12716,71	18554,45	-12182,46
0,88	-12960,65	-12791,79	16010,47	-12229,21
0,89	-13012,38	-12862,78	13620,43	-12274,16
0,9	-13061,54	-12929,85	11379,02	-12317,3
0,91	-13108,23	-12993,21	9280,62	-12358,72
0,92	-13152,56	-13052,96	7319,63	-12398,48
0,93	-13194,64	-13109,39	5490,11	-12436,63
0,94	-13234,55	-13162,51	3786,41	-12473,22
0,95	-13272,41	-13212,59	2202,7	-12508,33
0,96	-13308,31	-13259,74	733,13	-12541,99
0,97	-13342,34	-13304,1	-627,85	-12574,22
0,98	-13374,58	-13345,85	-1885,96	-12605,17
0,99	-13405,14	-13385,06	-3046,62	-12634,8
1	-13434,08	-13421,94	-4115,18	-12663,18

Tabla 93: Resultados de la evaluación del experimento 7 B utilizando como sistema de apuestas el criterio de Kelly.